STM JOURNALS

Journal of
Electronic Design Technology

ISSN: 2229-6980 (Online)
ISSN: 2321-4228 (Print)
Volume 17, Issue 1, 2026
January–April
DOI (Journal): 10.37591/JoEDT

https://journals.stmjournals.com/joedt

Electronic Design Technology: From Circuits to Smart Systems

V. Basil Hans*

Abstract

Electronic design technology has changed a lot over the years. It has gone from making simple circuits to making smart, complicated systems that run modern life. This article looks at how electronic design methods have changed over time, focusing on important improvements in tools, materials, and integrated systems. It looks at how new technologies like computer-aided design (CAD), embedded systems, and automation have made product creation faster, more accurate, and more scalable. The report also talks about how smart technologies like the Internet of Things (IoT), artificial intelligence, and miniaturization are becoming more important in making the next generation of electronic gadgets. Electronic design technology keeps pushing the envelope in fields including healthcare, communication, automotive, and consumer electronics by combining traditional circuit design with intelligent system integration. Furthermore, the growing use of cloud-based design platforms, digital twins, and sophisticated simulation methods is hastening innovation and shortening the time required to bring complex electronic systems to market. These advancements are promoting a design environment that is more collaborative and driven by data. The article ends by talking about current problems and future chances, stressing how important it is to use sustainable design, work with people from other fields, and keep up with new technology in a world that is becoming more linked.

Keywords: Electronic design technology, embedded systems, the Internet of Things (IoT), computer-aided design (CAD), and smart systems

INTRODUCTION

The need for electronic design automation (EDA) has grown as technology has changed and designs have become more complicated. EDA helps engineers design, test, and develop these systems more quickly and reliably. The technique has become more advanced, going from helping to create discrete circuits to smart systems architecture (SSA). This allows designers to use different technologies while satisfying requirements such as cost, energy use, performance, and time to market [1].

The electronic design includes system-level modelling, co-design of hardware and software, and selection of a platform for digital signal processing systems. System-level design is very important in the digital world for obtaining a characteristic that is proper by design. In recent years, algorithmic-level early-stage modelling, system-level simulation, high-level synthesis, and application-specific integrated circuit (ASIC) synthesis have all been studied to help designers understand what a system needs and speed up the design process. In the EDA toolset for these methods, techniques for physical design, formal verification, synchronous communication, high-level synthesis, scheduling, and architecture modelling have all changed over time [2].

***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 28, 2026
Accepted Date: March 31, 2026
Published Date: April 02, 2026

Citation: V. Basil Hans. Electronic Design Technology: From Circuits to Smart Systems. Journal of Electronic Design Technology. 2026; 17(1): 25–36p.

BASIC PRINCIPLES OF ELECTRONIC DESIGN

The microelectronic world became much more advanced when Lee De Forest invented the vacuum tube, and John Bardeen, Walter Brattain, and William Shockley invented the transistor. In September 1958, Jack Kilby built the first integrated circuit at Texas Instruments. This led to rapid progress that has continued ever since. Around the same time, Robert Noyce and Fairchild Semiconductor used a different method to create a silicon integrated circuit. These integrated circuits eventually led to silicon integrated circuit chips.

Computer-aided design methods, which were first used in the early 1970s to create very large scale integration (VLSI) of general-purpose semiconductor integrated circuits, were swiftly changed to ASICs in the mid-1980s. Algorithms were created to define, simulate, optimize, separate into hardware and software, plan, lay out, and check the entire design. Consequently, the semiconductor design community quickly and successfully moved to the difficult, time-consuming, and expensive task of ASIC design.

Basic physics and sophisticated mathematics are very important for all electrical and electronic engineering. However, at lower levels, electronic circuit theory, modelling, and simulation are equally important on many design platforms. The performance of any system (analog, digital, radio frequency (RF), mixed-signal, photonics, etc.) depends significantly on how well the electronic circuits are chosen to go with discrete components or integrated circuits. In universities and research laboratories, the steps of circuit design, modelling and exploration, layout, and fabrication are generally repeated multiple times. As technology improves, the period between each cycle increases [1, 3].

A Brief History of Circuit Theory

Circuit theory is the basis for designing circuits and systems. It discusses how power supplies work and how electrical currents and voltages are related to each other using clear mathematical models. A constitutive equation is a mathematical formula that describes the voltage across a device or the current flowing through the device. The foundation of circuit theory is based on the perfect voltage and current sources of passive and active components. This leads to the characterization of the circuit conditions set by Kirchhoff's equations and the behavior of capacitors, resistors, and inductors [4]. People usually think about circuit theory when they think of electronic systems because these systems form the basis of modern engineering. The field has existed for more than ten years, and during that time, important circuits and systems have been studied, documented, and modelled repeatedly. This is why it has been so successful in recent years.

Subsequently, the analysis becomes increasingly accurate. Numerous innovative strategies and methods have been developed in circuit theory. These methods facilitate the separation of components based on the concepts of constitutive equations. Such distinct delineations facilitate enhanced cognitive and operational efficiency for circuit and system designers along the historical progression of theory, modelling, and simulation within the microelectronic realm [2].

Basic Design Methods

There are many different methods and techniques used in electrical design to help people come up with ideas for and build systems and electronic circuits. There are usually two primary types of these methods: top-down and bottom-up. The top-down method is commonly used to look at a wide, abstract design space, whereas the bottom-up method is more focused on putting together parts chosen from predefined libraries or building more specific micro-architectures or circuit stages. Different levels of abstraction that define the system's behavior, architecture, and circuits are one of the most important parts of the top-down method [2]. Abstraction levels are very important during specification because they show the bigger design space, model the important features that show how complicated systems work, and let you run simulations before putting the hardware in place.

The design space defines a set of design choices that satisfy the criteria established at various levels of abstraction. This makes it more likely that the best option(s) will be chosen, even if it means putting together the more thorough implementation. Even while design exploration is generally thought of as going from the top-down or the bottom-up, it is now more common to look at two layers of abstraction at a time instead of all of them or blocks. Chip exploration, or layout micro-architecture research, looks at how important parts are connected to each other. Circuit exploration, on the other hand, looks at the whole block [3]. Design space exploration remains a dynamic and demanding field of research, focusing on the selection of design alternatives for specifications articulated at multiple levels of abstraction. The three main ways to use modelling and simulation to test electro-thermo-mechanical systems are analytical, symbolic, and numerical.

Tools for Modelling and Simulation

Integrated circuits have brought about a time of rapid technological progress; however, the design problems that arise with the fabrication of analog, RF, and mixed-signal circuits are still very real today. As frequency and modulation techniques have grown to include hundreds of megahertz and even multi-gigahertz ranges, circuit parts have moved into the microwave frequency range, and chip sizes have been reduced to less than a micrometer. In addition, emerging technologies have enabled chip makers to compete fiercely with each other for market share by using advanced signal processing methods. Therefore, as the need for circuit functionality and capability grows, design automation remains important for top-tier companies.

In the last 30 years, academic and industrial research has developed many useful and advanced tools for system design, synthesis, and verification, which have made the VLSI design process easier. However, traditional EDA tools cannot handle all circuit specifications simultaneously because technology has made circuits much more complicated. Most existing tools only consider one or a few design parameters, including delay, power use, or chip area. Other metrics were not considered. However, when circuit sizes and technology nodes become smaller, the original set of design parameters often does not agree with each other. For example, making a circuit smaller lowers its resistance, which speeds it up. However, smaller circuits also allow more current and power to leak out, which is a significant problem for both design metrics [5].

THE GROWTH OF ELECTRONIC DESIGN TECHNOLOGY

The technology available to designers has led to the development of EDA tools and methods. The technical changes that have occurred over the past 60 years have greatly increased the need and demand for these kinds of electronic structures, which shape these systems. In less than 30 years, the capacity of unintegrated or stand-alone designs has changed from making radio filters to making gigahertz digital signal processor filters. This is strong evidence that these technologies are being used to make EDA more complicated.

Scientists are now in a scenario where creation and design are becoming easier, even though the complexity continues to grow. This is because the automation and approach of the design tools are getting better all the time.

The fundamental concepts, methodologies, and frameworks established to promote the advancement of EDA for electronic design automation instruments are applicable to various prevalent and expanding research domains. Globalization and the necessity for design rules mean that these issues must be addressed [6].

From Separate Circuits to Integrated Circuits

In the late 1950s, silicon planar and hybrid circuit technologies made it possible to combine separate transistors, resistors, and capacitors into one unit. The first steps toward integration were tiny groups of hybrid circuits (composed of different types of technology) and silicon monolithic analog integrated circuits [2]. All operational amplifiers, voltage comparators, and oscillators were integrated into the analog building blocks. There was considerable interest in the design of integrated circuits that anyone could use, especially analog circuits.

The integration of discrete circuits kept the industry going until 1966, when dynamic random-access memory (DRAM) was introduced, and 1968, when the gate array came out and added a lot of digital and switching behavior. In 1965, silicon gate technology and in 1968, double-level metal technology made it possible to put more than 1000 MOSFETs on a single chip [3]. In 1970, discrete circuit integration of analog functions with comparable technology (such as low-current Emitter-Coupled Logic) began. This became a common technique. The gate array, which was produced in the early 1970s, was a big step forward in the integration of microelectronic systems. Around 1975, the standard microprogrammed control unit of the MS-1103 architecture had about 10^5 transistors. The number of transistors kept going up. In the midst of a single analog cell, it was possible to combine about 100 elements. At the same time, the number of complementary metal-oxide-semiconductor (CMOS) transistors in a mixed-technology chip grew a lot.

The Rise of VLSI and ASICs

As integration has improved, it is now possible to fabricate VLSI systems on chips. There are tens of millions to billions of electronic parts (transistors, resistors, capacitors, etc.) on the VLSI chip that can perform a certain job. As designs become larger and more complicated, it becomes harder to develop VLSI circuits, which makes problems more difficult to solve [7]. The productivity of design has not kept pace with the rapid growth of design complexity. The design method is basically the same whether chips have a few thousand transistors or billions. The biggest difference between the two designs is the degree of abstraction of the design and the appearance of the design domain. There are two main types of VLSI design: ASICs and circuits that do not use ASICs. A system-on-chip (SoC) is a circuit designed for a specific application. An ordinary ASIC circuit can be used for many or just a few applications [8].

Heterogeneous Integration and System-On-Chip

At the 2001 IEEE Custom Integrated Circuits Conference (CISTM), a SoC design was announced. It includes both mechanical and electronic parts, both analog and digital. To meet the current needs and new standards in EDA, the IEEE established a special SoC interest group and wrote rules for related technologies and standards [6]. Heterogeneous integration, which works with SoC methods, combines different microsystems, such as integrated circuits (IC), micro-electromechanical systems (MEMS), and micro-optical systems, on a substrate or in a package to create a complete system. It allows for a variety of choices in terms of size, technology, materials, and functions; it allows one to focus on designing systems instead of subsystems; it takes advantage of adding functional blocks to improve features or capabilities; and it makes it easier to replace old technologies. SoCs and heterogeneous integration are large and complicated systems that include many functions/modules. They have different physical, electrical, and thermal properties and are connected across different technologies by imperfect, high-performance interconnections that may not function well.

THE MOVE TO SMART SYSTEMS

Significant changes have occurred in embedded real-time systems since the 1990s. This progress was made possible by several factors, such as better global connectivity, lower prices for computer platforms, the early creation of standardized code development languages, and the availability of several operating systems. Short descriptions of the type or class can help clarify the dependencies, as the terms “real-time system” and “real-time environment” can have various meanings. The most evident factor is whether a system is hard or soft real-time. Incremental embedded systems usually operate under hard real-time circumstances. You must follow the timing rules exactly; otherwise, the system will fail. The problem is more complicated when the target functions, timing restrictions, and resource capabilities change with time, but the timing constraints remain the same. The first two set a mandatory real-time limit, the third and fourth set parameterized—flexible—limits, and the last class does not have any mandatory requirements [1].

One of the tasks of these embedded systems is to handle the huge amounts of data that come in structured, continuous streams from sensor and transducer systems that are becoming more common. An increasing number of people are using high-fidelity sensors, including microphones and cameras, scoped angular fields, thermal, radar, infrared, and position sensitivity, as well as accelerometers, gyros, and RF rendezvous. An access-processing-store post-processing transmission deployment tree is in effect from the outset [2].

Real-Time Computing and Embedded Systems

When people talk about embedded systems or devices, they usually mean computers or systems that are an important part of a bigger electronic system. Early embedded systems used microcontrollers that were made for a specific purpose, and timers were typically used to make sure that the system behaved in a predictable way and to meet simple timing requirements. Modern embedded systems now use general-purpose central processing units (CPUs), which means they cannot provide real-time (or deterministic) behavior anymore. The change from microcontroller (MCU) to microprocessor (MPU) is thought to have a big effect on how these embedded systems develop. Embedded computer systems are characterized by a hardware–software combination tailored to execute dedicated functions [9]. The research indicates that early computing systems significantly influenced the design of embedded computing systems. Embedded computing systems are in a lot of things we use every day, like cars, smart appliances, wearable tech, phones, laptops, and more. Embedded computing systems require knowledge of both hardware and software, as well as hardware–software co-design, because they are based on computer science and electrical engineering. Embedded System Education has changed over the past twenty years to meet the needs of different fields and industries. The development of embedded systems began with early real-time computers and has advanced through embedded computer systems to the pervasive embedded devices such as the Internet of Things (IoT). Embedded System instruction has become an essential aspect of engineering education.

The Internet of Things and Edge Computing

The shift of artificial intelligence (AI) from centralized cloud systems to edge computing has made it possible to use it in more places, especially in IoT. However, moving AI algorithms from the cloud to the edge raises new architectural problems. There are many possible edge architectures, such as edge-only, cloud-edge, and sensor-edge architectures. For each edge architecture choice, the method by which data is handled and the need for low-latency processing must be carefully considered [10]. Recent improvements in processing and communication technology have set a standard for designing IoT systems that attempt to make the best use of energy and hardware resources. Edge computing, which makes it possible to process data in real time close to the source instead of sending it to the cloud, and wireless sensor networks, which adjust resources at both the communication and computation levels, are two important components of this paradigm [11].

AI in Hardware Design

The rapid progress of AI has changed several fields, creating new opportunities for people to be creative and come up with new ideas [12]. AI is used in hardware design to solve problems, including shorter design cycles, reliability difficulties, more complexity, and more variety [1]. The quality of the model determines how well AI and hardware work; therefore, new advances in the electronic design of ICs and algorithmic improvements in AI go hand in hand. AI is about to change the hardware industry by improving design, optimization, verification, and even co-design. AI can help determine activations, parameters, topologies, and architectures, which will make your designs crisper and more accurate. Chip designers, electronics engineers, system architects, and hardware–software designers are beginning to use AI and its components in their work every day. This expansion is aided by the rise of devices that speed up AI calculations and make learning on the device easier. One goal for designers is to achieve the best performance from each watt. This will allow for more accurate recognition, less delay, higher efficiency, more options for tackling difficult issues, and many other benefits to be realized. There are more and more data centers around the world that include chips that can perform many functions with a small amount of power. These chips are powerful and assist with the design.

With the rise of GPUs, DSPs (digital signal processors), and dedicated processors for speed, AI technology has come a long way. These advances provide designers with new ways to improve the design process. There are still problems with defining AI in the context of electronic design and determining the AI models. Different AI models help design hardware by acting as system architects or by facilitating the development of hardware and software together. There are many places where AI is appropriate, but there are currently no rules in place. Checks must be performed to ensure that the limits are acceptable. Some important factors that affect usefulness, such as resource occupancy, frontiers, competitive status, backward compatibility, and optimum circumstances, remain unclear.

The seamless integration of AI into hardware design represents the most recent advancement in the development of completely autonomous intelligent devices. AI now allows computers to create their own mathematical models and algorithms and write code for hardware implementations. The creation of more advanced hardware is still important and useful, and additional progress is expected to lead to even faster progress.

Design Approaches for Contemporary Systems

Modern systems increasingly necessitate design techniques that effectively tackle functional partitioning and hardware–software interactions, as advocated by hardware–software co-design. Model-based systems engineering can formalize system requirements; however, the lack of formal semantics may cause ambiguity. Verification, validation, and certification ensure that systems fulfil specifications and follow the correct standards. They do this by using different design, testing, and simulation methods to examine safety, security, resilience, performance, and other factors [6].

Hardware–Software Co-Design

Smart systems sometimes use non-standard architecture; thus, it is important to consider the hardware–software split carefully. The goal of hardware–software co-design is to improve both hardware and software platforms simultaneously. This method raises many concerns, such as the tasks to be provided with the hardware and software parts. Co-design also includes co-simulation methods that allow one to examine different ways to split things up and see how they affect stage duration, latency, and line capacity.

Model-Based Systems Engineering

The need for more comprehensive requirements coverage and the growing complexity of systems have made model-based systems engineering popular. This approach formalizes system requirements and uses models that can be used to verify them. However, the focus on safety frequently prevents us from considering other important aspects, such as security, resilience, and performance. The absence of formal semantics sometimes generates ambiguities in model-based system designs, creating significant obstacles to the verification of essential features.

Co-Design of Hardware and Software

Programmable logic devices and microcontrollers have changed the way electronics are designed by making it possible to perform tasks in both hardware and software. This change has made it more difficult to distinguish between hardware and software. Consequently, decisions regarding how to split them are crucial for performance, cost, power use, and risk. Co-design solves this problem by connecting hardware and software development throughout the design process. A top-down co-design process captures high-level functionality for the early evaluation of different architectures, whereas the regular simulation of virtual prototypes with hardware–software interaction and co-simulation on accurate models promotes convergence to final implementations [13].

Engineering Systems Based on Models

A well-designed model shows how the system should work and checks to see if the program does what it says it will. Changes to this kind of model set off a chain reaction that affects system documentation, design methodologies, data-processing algorithms, and software. The same kinds of rules apply to how models, designs, and software change during the implementation phase [14]. Sufficient validation of models, designs, and software prior to implementation enables the staggered delivery of the associated validation report, effectively addressing the dilemma of verification coordination [15]. Models or partial models with stable content serve as the verification baseline for evaluating subsequent changes. Model transformation is the process of changing a hardware design to see if it will work. It controls how models at different levels of abstraction relate to each other. It is still very important to keep a dual-level description of machinery, and model transformation is still quite important. Arrow transformation or similar ideas show promise when it comes to designing for testing instead of verification. Adding model-based design rule checking (with a domain-specific language and compiler) to the design flow is another way to make engineering more coherent. Domain-specific languages make it easier for the behavior specifications of the system being created to work well with the electronic parts that make it up. This lets designers utilize the right models at every stage of the design process.

Verification, Validation, and Certification

Verification, validation, and certification (VVC) procedures and metrics for both hardware and software are essential yet frequently neglected aspects in the design of intelligent systems. Verification is the process of checking to see if a design accurately matches functional specifications or user needs. Validation, on the other hand, is the general word for scientifically showing that a system or its parts meet requirements set by stakeholders or users [16]. Certification is another important idea. It works with verification and validation to make sure that anything meets international or industry standards [17]. VVC has long been a major focus in embedded system design when it comes to software. However, the rise of hardware concepts like field-programmable gate arrays (FPGAs) and ASICs has made hardware certification a top priority for safety-critical standards.

Because SoCs can combine several different systems on one chip, their certification has become more difficult. Now, each aspect of SoC must be checked, validated, and certified independently. Because general-purpose processors are built in, it is especially vital to meet user-defined and system-level criteria. Therefore, VVC is now an integral part of designing smart systems. DO-254 is one of several accepted standards for the VVC of hardware. Internationally important safety-critical applications in avionics, such as DO-178, have led to the creation of equivalent rules for hardware.

TECHNOLOGIES THAT POWER SMART SYSTEMS

As smart systems become more complicated, there is a greater requirement for the right kind of sensors, actuators, calibration tools, and control loops. The focus on data-centric architecture instead of a process-centric one affects the integration of chips into systems. In addition, power management systems that connect many regulators across a range of topologies, from generic low-dropout or switching-grid controllers to ASICs, must be energy-efficient, have voltage domains, and be reliable. Protocols control how the SoC infrastructure communicates with each other and how chips and boards communicate with each other. The type of network topology you choose, such as buses, point-to-point links, or meshes, can affect performance while still satisfying requirements such as bandwidth, latency, fault tolerance, and access priority. Security problems can lead people to use heterogeneous networks. Quality of service (QoS) standards include data integrity, guaranteed latency, and bandwidth monitoring for real-time communication. Additional requirements may include “plug-and-play” features that allow the addition of new parts to the system without disrupting the existing ones [6].

Actuation and Sensor Robotics

Robot sensors and actuators are ubiquitous. They characterize their surroundings and allow robots to interact with the world in smart ways to get things done. Many sensing modes check for changes in the environment, such as light, sound, position, or touch. As robotics design systems become smarter, we need transducers that can provide the right information. Sensors are devices that measure changes over time, such as temperature. Actuators, on the other hand, are used to make mechanical changes (such as motors, piezo devices, and valves) or electrical changes (such as electronic switches, solid-state relays, and lighting).

There are many different types of smart sensors, such as fiber-optic, ultrasonic, and microwave sensors. Smart sensors also use sound waves, capacitive, electromagnetic, electromagnetic, gas, image, infrared, mass, mechanical, magnetic, optical, radar, tactile, temperature, and vibration [18]. Standards exist for sensor systems used in smart buildings, laboratories, factories, and offices to monitor the environment. An environmental monitoring system comprises one or more sensors, data acquisition components, data conditioning systems, process equipment status systems, human-computer interface devices, and computers.

Managing Power and Collecting Energy

Power management circuits ensure that gadgets receive power at the correct voltage. When, where, and how energy is available, harvesting it can either make batteries last longer or eliminate them altogether. Energy harvesting technologies can work with batteries to extend the life of gadgets that run on batteries, such as smartwatches. This opens up new areas for energy harvesting applications, such as logistics, door plates, and shelf labels [19]. Energy harvesting can be used in a wide range of devices with power management circuits that operate from nanowatts to watts [20].

Fabric for Networking and Communication

The biggest problem with interconnect fabrics is moving data from one spot on a chip to another with the least amount of delay and energy use, and the most reliability. Traditional VLSI designs employ separate wires to connect resources; however, as things become more complicated and smaller, new methods are required. Miniaturization does not improve the performance of wires because global wires slow down the process, making it harder to scale. Recent advancements encompass 3D architecture and network-on-chip (NoC) designs, which separate communication from computing units and improve the adaptability of the devices. NoCs use packet-oriented traffic with programmable switch blocks but introduce area and energy overheads, depending on the connectivity, complexity, and number of switches [21].

MAKING THINGS AND THE ECOSYSTEM FOR ELECTRONIC DESIGN

The design of modern electronics has come a long way since the 1950s, when it was only circuit design. The first steps in building electronic systems required the use of devices such as bipolar junction transistors, field-effect transistors, and operational amplifiers to analyze and synthesize circuits. Tools were created to help designers model, simulate, and synthesize circuits more easily. New tools and methods have been made available to assist electronic designers as the levels of integration and costs of development have increased. The 1980s were a time when VLSI design approaches, algorithms, and software tools based on framework software, such as individual computer-aided design packages and EDA platforms, became popular. The IoT presents distinct difficulties with current tools, necessitating the development of novel computational models, co-design approaches, and an expanded array of communication and processing techniques for smart system design. These changes have led to the creation and growth of electrical design technologies, which have received increasing attention from both academia and industry [6].

Processes and Yield for Making

When making things with medium- and deep-submicron technology nodes, the lines that are deposited are 1 μm wide or less. Also, problems with the fabrication process and the atmosphere that are not perfect can make it hard to get good performance and yield. For example, 90 nm CMOS technology has a σ-σ variance (for example, fast and slow corners) that can cause a 40% change in leakage for nMOS. This is because the drain current changes by the same amount, but in an unusual way [2]. So, changes need to be made at different stages of the design process. Also, when just post-layout optimization is used, with distributed capacitance and leakage area as goals and parasitic capacitance and hold time set-up transistors taken into account, a yield effectiveness ratio of 86% is reached.

The inherent random doping and charge interface cause the threshold voltage to vary widely in 90 nm technologies. The absolute value of the on current changed from 76–106.2 μA to 136.8–179 μA because of the design, the way it was made, and the environment. These significant discrepancies in the circuit simulation process necessitate the redesign and augmentation of the design cycle, both on-chip and off-chip, for analog circuits [22].

Standards, Toolchains, and Reuse

The design of electronic systems and circuits is crucial to the complex electronic economy of today. To perform system-level design, many tools that work together in a comprehensive design flow are required. The EDA community has created design standards for programming languages, graphics representations, intermediate formats, circuit and system specifications, design environments, hardware description languages (HDLs), verification, layouts, performance, simulation, power, and test generation. These standards are intended to encourage the automation of the design of analog, digital, and mixed-signal systems. Soft IP (Intellectual Property standard-compliant) and Hard IP (physical implementation standard-compliant) cores [6] further assist these design standards. Therefore, the design process remains platform-independent, which means that different design flows and programmed design programs can work together.

Working Together in Education, Research, and Business

The changing world of electronic design technology requires many talented people to work in it. Previously, educators and researchers trained engineers who improved embedded computing systems, electronics, and semiconductor technologies, leading to the growth of a multi-trillion-dollar global business [9]. Embedded systems education has difficulty keeping up with the rapid changes in technology, and many of the courses that are already available do not have sufficient relevant material. New technologies such as IoT, AI, self-driving cars, and augmented and virtual reality are always changing and moving too quickly for schools to keep up with. As a result, partnerships between educational research and industry are moving forward with both educational and industrial prospects to create and shape the future generation of curricula. A related problem is that we need a new generation of engineers who can work across disciplines and possess the skills required to design and build embedded computer systems throughout their lives.

CONSEQUENCES AND OBSTACLES

Electronic systems are ubiquitous and are an important part of modern culture. However, there are many problems with the design, production, and use of IC and electronic systems. These problems can cause malfunctions, dangerous behaviors, or even damage to parts and systems. Foundation models and the associated technologies that go along with them can facilitate the development of both new and existing electronic systems [1]. Electronic systems are an integral part of modern life. These systems are now smaller, faster, and stronger. For society to run well, it is important that things are always available, work all the time, and are safe from bad external conditions and intrusions. The issues of reliability, security, and privacy that arose during the creation of IC and electronic systems need to be examined.

The lifespan and energy costs of the entire technology stack are a significant threat to the world today and in the future. The extraction, manufacturing, shipping, and various processing steps of raw materials used to produce electronic equipment significantly affect the environment. As more people use AI, the need for more computing power grows, which means that energy use increases. The worsening state of the environment and the growing need for more computing capacity for AI systems put the existence and quality of life of future generations at risk if we do not find good and timely answers. Therefore, society is in a delicate balance between living a healthy life and being able to continue [2].

Dependability, Safety, and Privacy

Modern societies depend on a complex network of cyber-physical systems, including smart sensors, actuators, and computers built into other devices. The threat from malicious people who want to break these systems goes beyond software to include hardware. Currently, an attack on an electronic or cyber-physical system can cause problems, damage the environment, cause loss of reputation, and even cause bodily harm. The growing number of assaults is mostly due to top-down design flows, design cycles that take longer than intended, and insufficient knowledge about threats over the design's full life cycle [1].

The third thesis maintains that security remains a primary priority while presenting additional problems. Side-channel and fault-injection attacks on processors can lead to information leaks and other problems. The Equifax breach and the Spectre and Meltdown vulnerabilities that affect billions of devices demonstrate this. These threats are aimed at both digital and analog components. However, the increasing number of system parts makes the design more complicated, making it difficult to determine the system's longevity, size, and carbon production throughout its entire life cycle. However, IoT is increasing the need for and replacement cycles of different devices. In addition, the gaseous emissions from manufacturing are often worse than those from the actual use of the element in question [23].

Ethical and Environmental Issues

The design of sustainable integrated electronic systems is becoming increasingly important. Innovators, managers, and regulators must embrace a lifecycle view to assess and enhance the diverse direct and indirect sustainability consequences of system designs [24]. System inventors should establish responsibility for their involvement in political, economic, ethical, and sustainability-related matters [25]. The ever-changing and more complicated nature of scientific and engineering systems makes it difficult to determine who is morally responsible for what.

Effects On the Economy and the Workforce

Three major factors that have a significant effect on the economy are the level of EDA tool adoption, their complexity, and the level of integration accomplished. These are the factors that are changing design technology. The chip market has concentrated on mass production. It has become linked to politics and has caused prices to change significantly because of economic cycles and the state of the world [26].

CONCLUSION

Electronic design has come a long way since it first started making simple circuits. Now, it makes quite complicated systems that can do advanced tasks on their own. The basic rules for designing circuits were set down in the early 1800s, and the equipment and methods needed to do so quickly improved in the second half of the 1900s. Even though electronic design technology is always getting better and the design ecosystem is getting bigger, progress has slowed down because electronic designs are now so complicated that they are hard to work with. It is getting harder and more expensive to build new circuits and systems. Design research is entering a new phase with new needs and limits, new chances for design exploration, and new study themes and problems. The current change is called "the Rise of Smart Electronic Design" or "the Rise of Smart Systems Design." The rise of smart systems design means moving from circuit to system and from implementation to specification. A set of new requirements and limits, real-time, power, physical, data, process, and economics, has come up that has a big effect on parallel and distributed electrical design. Simultaneously, novel opportunities and a paradigm shift in design exploration have emerged, allowing for the integration of AI into the design process. To solve the main difficulties with modern design and to guide electronic design research in schools and businesses, we need to know what motivated this change in the past and what the main factors were.

REFERENCES

1. Knechtel J, Kavun EB, Regazzoni F, Heuser A, Chattopadhyay A, Mukhopadhyay D, et al. Towards secure composition of integrated circuits and electronic systems: on the role of EDA. 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France. 2020. p. 508–513. doi:10.23919/DATE48585.2020.9116483.

2. Pribyl W. Integrated smart power circuits technology, design and application. ESSCIRC '96: Proceedings of the 22nd European Solid-State Circuits Conference, Neuchatel, Switzerland, 1996. p. 19–26.
3. Cirstea M, Benkrid K, Dinu A, Ghiriti R, Petreus D. Digital electronic system-on-chip design: methodologies, tools, evolution, and trends. Micromachines. 2024;15(2):247. doi:10.3390/mi15020247.
4. Lundstrom M. Computational electronics for the 21st century: reflections on the past, present, and future. 2015 45th European Solid State Device Research Conference (ESSDERC), Graz, Austria. 2015. p. 36–39. doi:10.1109/ESSDERC.2015.7324707.
5. Chang JS, Facchetti AF, Reuss R. A circuits and systems perspective of organic/printed electronics: review, challenges, and contemporary and emerging design approaches. IEEE J Emerg Sel Top Circuits Syst. 2017;7(1):7–26. doi:10.1109/JETCAS.2017.2673863.
6. Martin G, Chang H. System-on-chip design. 2001 4th International Conference on ASIC Proceedings (ASICON 2001) (Cat. No.01TH8549), Shanghai, China. 2001. p. 12–17. doi:10.1109/ICASIC.2001.982487.
7. Liu HY. Supervised design-space exploration [thesis]. New York (NY): Columbia University; 2015.
8. Harshitha NB, Praveen Kumar YG, Kurian MZ. An introduction to universal verification methodology for the digital design of integrated circuits: a review. 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India. 2021. p. 1710–1713. doi:10.1109/ICAIS50930.2021.9396034.
9. Pasricha S. Embedded systems education in the 2020s: challenges, reflections, and future directions. Proceedings of the Great Lakes Symposium on VLSI 2022 (GLSVLSI '22); 2022 Jun; Irvine, CA, USA. New York (NY): Association for Computing Machinery; 2022. p.519–524. doi:10.1145/3526241.3530348.
10. Cárdenas R, Arroba P, Risco Martín JL. Bringing AI to the edge: a formal M&S specification to deploy effective IoT architectures. J Simul. 2022;16(5):494–511. doi:10.1080/17477778.2020.1863755.
11. Sheng Z, Tian D, Leung VCM. Toward an energy and resource efficient internet of things: a design principle combining computation, communications, and protocols. IEEE Commun Mag. 2018;56(7):89–95. doi:10.1109/MCOM.2018.1700536.
12. Wheeldon A, Shafik R, Rahman T, Lei J, Yakovlev A, Granmo OC. Learning automata based energy-efficient AI hardware design for IoT applications. Philos Trans A Math Phys Eng Sci. 2020;378:20190593. doi:10.1098/rsta.2019.0593.
13. Beetner D, Pottinger H, Mitchell K. Laboratories teaching concepts in microcontrollers and hardware-software co-design. 30th Annual Frontiers in Education Conference. Building on A Century of Progress in Engineering Education. Conference Proceedings (IEEE Cat. No.00CH37135), Kansas City, MO, USA. 2000. vol. 2. pp. S1C/1–S1C/5. doi:10.1109/FIE.2000.896613.
14. Wheaton JS, Herber DR. Seamless digital engineering: a grand challenge driven by needs. In: AIAA SCITECH 2024 Forum; 2024 Jan 8–12; Orlando, FL, USA. Reston (VA): American Institute of Aeronautics and Astronautics; 2024. AIAA 2024-1053. doi:10.2514/6.2024-1053.
15. Varadharajan SK, Nallasamy V. Low power VLSI circuits design strategies and methodologies: a literature review. 2017 Conference on Emerging Devices and Smart Systems (ICEDSS), Mallasamudram, India. 2017. p. 245–251. doi:10.1109/ICEDSS.2017.8073688.
16. Srivastava A, Gupta A, Anand R. Optimized smart system for transportation using RFID technology. Math Eng Sci Aerosp. 2021;12(4). Available from: https://nonlinearstudies.com/index.php/mesa/article/view/2749
17. Kornecki AJ, Zalewski J. Hardware certification for real-time safety-critical systems: state of the art. Annu Rev Control. 2010;34(1):163–174. doi:10.1016/j.arcontrol.2009.12.003.
18. Najafi K. Smart sensors. J Micromech Microeng. 1991;1(2):86–102. doi:10.1088/0960-1317/1/2/002.
19. Teodorescu HN, Jain LC, editors. Intelligent Systems and Technologies in Rehabilitation Engineering. Boca Raton (FL): CRC Press; 2000.

20. Gao RX, Suryavanshi A. BIT for intelligent system design and condition monitoring. IEEE Trans Instrum Meas. 2002;51(5):1061–1067. doi:10.1109/TIM.2002.807796.
21. Teuscher C. Nature-inspired interconnects for self-assembled large-scale network-on-chip designs. Chaos. 2007;17(2):026106. doi:10.1063/1.2740566.
22. Jackson E, Sun M, Kubota T, Takanashi K, Hirohata A. Chemical and structural analysis on magnetic tunnel junctions using a decelerated scanning electron beam. Sci Rep. 2018;8(1):7585. doi:10.1038/s41598-018-25638-8.
23. Singh G, Kaur A. Sustainable digital circuit design: asynchronous counter on Kintex and Virtex FPGA platforms. 2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India. 2024. p. 97–101. doi:10.1109/ICSCSS60660.2024.10625483.
24. Mullen E, Morris MA. Green nanofabrication opportunities in the semiconductor industry: a life cycle perspective. Nanomaterials. 2021;11(5):1085. doi:10.3390/nano11051085.
25. Fiore E. Design ethics in socio-technical systems: addressing the ethics of connected appliances. In: Relating Systems Thinking and Design Symposium (RSD5); 2016 Oct 13–15; Toronto, Canada. Toronto (CA): OCAD University Open Research Repository; 2016. Available from: https://openresearch.ocadu.ca/id/eprint/1941/
26. Goldstein H. Community impacts of new industrial development. Carolina Planning J. 1985;11(2):29–39. Doi:10.17615/xaah-3n49.