

---

---

# Preface

---

---

Since the fourth edition of *Digital Design*, the commercial availability of devices using digital technology to receive, manipulate, and transmit information seems to have exploded. Cell phones and handheld devices of various kinds offer new, competing features almost daily. Underneath the attractive graphical user interface of all of these devices sits a digital system that processes data in a binary format. The theoretical foundations of these systems have not changed much; indeed, one could argue that the stability of the core theory, coupled with modern design tools, has promoted the widespread response of manufacturers to the opportunities of the marketplace. Consequently, our refinement of our text has been guided by the need to equip our graduates with a solid understanding of digital machines and to introduce them to the methodology of modern design.

This edition of *Digital Design* builds on the previous four editions, and the feedback of the team of reviewers who helped set a direction for our presentation. The focus of the text has been sharpened to more closely reflect the content of a foundation course in digital design and the mainstream technology of today's digital systems: CMOS circuits. The intended audience is broad, embracing students of computer science, computer engineering, and electrical engineering. The key elements that the book focuses include (1) Boolean logic, (2) logic gates used by designers, (3) synchronous finite state machines, and (4) datapath controller design—all from a perspective of designing digital systems. This focus led to elimination of material more suited for a course in electronics. So the reader will not find here content for asynchronous machines or descriptions of bipolar transistors. Additionally, the widespread availability of web-based ancillary material prompted us to limit our discussion of field programmable gate arrays (FPGAs) to an introduction of devices offered by only one manufacturer, rather than two. Today's designers rely heavily on hardware description languages