

Episode 0.0 – Prologue

Welcome to the Geek Author series on Computer Organization and Design Fundamentals. I'm David Tarnoff, and I've been teaching Computer Organization, Computer Architecture, Digital Design, and Embedded System Design for over 25 years. This episode is going to be a brief introduction to the series and, well, to me.

Before starting my teaching career, I designed motherboards and BIOSes in industry. When I found myself at the front of a classroom filled with computer science students, I have to admit that I didn't quite know where to begin. I was an electrical engineer after all, not a computer scientist. Heck, I'd never even heard the term, "Computer Organization." Why is it important to organize computers? Would label makers be involved? Was Feng Shui important?

It turns out that what computer scientists call computer organization is what I learned in my EE curriculum as, "Digital Design."

With that conundrum out of the way, my next task was to figure out a topic list. The textbook used by my predecessors assumed the reader had had some experience with electronics. The author went so far as to use screen captures from oscilloscopes and other test equipment to describe circuit properties. I soon found that this was a bad assumption to make when it came to students of computer science.

So to provide a lifeline to my floundering students, I began writing supplementary notes and posting them to my course web site. Over the years, the notes matured until at one point, the students stopped buying the required textbook and simply studied from the online notes. When the online notes were discovered by search engines, I began receiving messages from other instructors asking if they could link to my notes. The answer was obvious: of course!

The online notes provided a wonderful opportunity. Instead of requiring a textbook for my course, I could ask my students to purchase hardware or software to supplement the university's laboratory equipment. This could include anything from test equipment to circuit components. I hoped that by enhancing the hands-on portion of the course, I improved each student's chance to learn and retain the material.

As the on-line notes became more popular, I began receiving comments from students about dried up printer cartridges and the cost of reams of paper. Note that this was back in the days when people read from paper and not from the screen of a smart phone. I once found a student searching the recycled paper bin for scrap paper on which to print my notes. Even our campus technology group had begun to suggest I was one of the causes for the overuse of campus printers. That was when I decided to compile my web notes into a book.

In 2005, I self-published the first edition of Computer Organization and Design Fundamentals. I made the pdf's free to anyone who had a desire to learn about the digital concepts of a computer. I tried addressing topics such as analog to digital conversion, CRC's, and memory organization using practical terms and examples instead of the purely theoretical or technical approaches favored by the engineers. Hopefully I succeeded. I never pretended to believe that my book could provide the reader with the background necessary to begin designing and building contemporary computer circuits. I did, however, believe that reading it would give people the tools to become better developers of software and

computer systems by understanding the tools for logic design and the organization of the computer's internals.

Over the past few years, I've noticed yet another change. Fewer and fewer students were reading the textbook. In fact, I gave about 50 of my students an informal survey where close to half of them admitted to not reading any of the textbook over the course of the semester. In 2005, I took advantage of the latest technology, so I guess almost 15 years later, it's time to take advantage of what's available to us now.

The purpose of this material hasn't changed. I still believe that an understanding of computer hardware can be applied to software design allowing for improved system performance. The material in this series can be used as a springboard to topics such as advanced computer architecture, embedded system design, network design, compiler design, or microprocessor design. The possibilities are endless.

The presentation of the material in this series will be similar to that of the original textbook. In the first third, we will discuss the mathematical foundation and design tools that address the digital nature of computers. This will include an introduction to the differences between the physical world and the digital world, how those differences affect the way the computer represents and manipulates data, and the use and design of digital logic and logic gates. In the second third, the fundamentals of the digital logic and design will be used to design common circuits such as binary adders, describe checksums and cyclic redundancy checks, network addressing, storage devices, and state machines. The final third will examine the top-level view of the computer. This will include a discussion of the memory hierarchy and its components, the components of a CPU, and maybe even a little about assembly language.

Effectively, this is the fourth edition of this book. That means I will be presenting new material and expanding existing material. Plans include additional sections on signed BCD arithmetic, biased-N notation, SPI and I2C protocols, and data compression, while sections such as IEEE-754 format, cache performance, and data representation will be updated.

And lastly, a disclaimer. The information in this book is based on my experience over many years of digital logic and embedded system design. I believe this information is correct, but I cannot accept responsibility or any liability whatsoever with regard to the application of any of the material presented in this book. In addition, the design tools presented here are meant to act as a foundation to future learning. I do not offer any warranty or guarantee toward products used or developed with material from this book. I also deny any liability arising out of the application of any tool or product discussed in this book. If the reader chooses to use the material in this book to implement a product, he or she shall indemnify and hold the author and any party involved in the publication of this book harmless against all claims, costs, or damages arising out of the direct or indirect application of the material.

Whew, I'm glad that's over with. Now let's get on with learning about computer organization. In our next episode, we will discuss the importance of hardware design to the software engineer along with some of the benefits of digital data over analog. Until then, remember that it is wrong to think that programming a computer is analogous to driving a car. No, browsing the Internet is analogous to driving a car. To effectively create computing solutions, however, you gotta look under the hood. But don't worry. While the scope of what makes a computer is immense, it's all just ones and zeros.