Fundamentals of Digital Circuit Design

Cezary Zieliński

Publisher: Wydawnictwo Naukowe PWN, Warsaw, 2003
(Published in Polish; contains 450 pages)

Objectives of the book:

The course is targeted at all computer science, electrical, electronics, telecommunications and mechanical engineers who require the understanding of the functioning and the design of logic circuits at different levels of their complexity. The motif for this lecture is to show what are the major components and how to design digital logic for real time systems (e.g. mechatronics, robotics, computer systems). The lecture starts with combinational logic and ends with the design of large digital systems. The circuits integrated to a higher extent or more complex systems are designed using the already introduced components. For instance, a D-type flip-flop is designed using gates, then in turn functional blocks, such as counters or registers, are designed using those flip-flops, and finally functional blocks are used to construct data processing and control systems. In this way no phase of integration is left out giving the reader a full understanding of the field. As a result the readers should be able to design circuits and systems using elements of any scale of integration according to the needs of a project. Moreover, different approaches to designing the same circuits or systems are explained. All the concepts are explained explicitly showing the design process by simple yet real engineering examples. The mathematical background is kept to the minimum necessary level (rudimentary Boolean algebra is utilised). The examples are chosen in such a way that the inner workings of computers are explained, starting at the gate level, traversing ever more complex modules, and ending with microprogrammed machines.

Table of contents:

2.1.10. Specification of logical functions
2.1.11. Fundamentals of mnemonic methods of minimising logical functions
2.1.12. Grays code
2.1.13. Karnaugh maps
2.1.14. Minimisation of logical functions using Karnaugh maps
2.1.15. Dont care values
2.1.16. Sum of Products normal form
2.1.17. Factorization
2.1.18. Prohibition
2.1.19. Factorization and prohibition combined
2.1.20. Product of Sums normal form
2.1.21. Quine and McCluskeys method of minimisation
2.1.22. Static hazard
2.1.23. Dynamic hazard
2.1.24. XOR gates

2.2. Complex combinational circuits

2.2.1. Iterative circuits with one-way flow of information between blocks
2.2.2. Iterative circuits with two-way flow of information between blocks
2.2.3. Cascade circuits
2.2.4. Combinational circuits utilising multiplexers
2.2.5. Combinational circuits utilising decoders
2.2.6. Demultiplexers
2.2.7. Read only memory
2.2.8. Application Specific Integrated Circuits

Design exercises

3. Synchronous circuits

3.1. General introduction

3.1.1. Problem formulation
3.1.2. Automaton state
3.1.3. Graphs
3.1.4. State transition and output table
3.1.5. Coding
3.1.6. D type flip-flop
3.1.7. Flip-flop excitation tables
3.1.8. Synchronous automaton realisation

3.2. Theoretical foundations

3.2.1. Synchronous automaton structures
3.2.2. Automaton
3.2.3. Equivalence of complete (fully specified) automaton states
3.2.4. Compatibility of incomplete (partially specified) automaton states
3.2.5. Formalisation of the minimisation algorithms
3.2.6. Realisation of a minimal automaton using D type flip-flops