1 Introductory concepts **1–1**
1.1 Switches. NOT operation . . . 1–1
1.2 Parallel connection of switches . . . 1–2
1.3 Serial connection of switches . . . 1–3
1.4 Exercise . . . 1–4

2 Numbers in digital systems **2–1**
2.1 Digits, numerals and numbers . . . 2–1
2.2 Multiplication and division by a power of the radix, r^m . . . 2–3
2.3 Basic arithmetic operations . . . 2–4
2.3.1 Addition in radix r system . . . 2–4
2.3.2 Multiplication by 10 in a binary system . . . 2–4
2.4 Radix conversion . . . 2–5
2.4.1 Method 1: Division by target radix in the source system . . . 2–5
2.4.2 Method 2: Multiplication by source radix in the target system . . . 2–7

3 Logic Gates and Boolean Algebra **3–1**
3.1 CMOS Technology . . . 3–1
3.2 Boolean Algebra and Logic Gates . . . 3–2
3.3 Timing diagrams . . . 3–4
3.4 Boolean Expressions and Logic Diagrams . . . 3–5
3.5 VHDL Hardware Description Language — example 1 . . . 3–7
3.6 Truth tables and Karnaugh Maps . . . 3–8
3.7 A 3-variable Karnaugh map . . . 3–9
3.8 Theorems of Boolean Algebra and their circuit interpretation . . . 3–9
3.9 All two-variable functions $y = F(b, a)$ . . . 3–16
3.10 NAND and NOR gates . . . 3–17

4 Canonical and standard forms **4–1**
4.1 Minterms. n-to-2^n Decoders . . . 4–2
4.2 The Sum-of-Minterms (SoM) canonical form of a logic function . . . 4–4
4.3 Decoder-based implementation of a logic function. Solution 1. . . . 4–5
4.4 Maxterms . . . 4–6
4.5 The Product-of-Maxterms (PoM) canonical form of a logic function . . . 4–7

4.6 More on Decoder-based implementation of a logic function. . . . 4–8
4.7 Standard forms . . . 4–9
4.8 NAND and NOR based implementations . . . 4–10

5 Gate-level minimization **5–1**
5.1 Principle of logic function minimization . . . 5–1
5.2 Karnaugh Maps . . . 5–2
5.3 A 2-variable Karnaugh map . . . 5–2
5.4 A 3-variable Karnaugh map . . . 5–4
5.5 A 4-variable Karnaugh map . . . 5–10

6 Combinational circuits **6–1**
6.1 Introductory concepts . . . 6–1
6.2 Example of a VHDL code for a 2-to-4 decoder . . . 6–3
6.3 Multiplexers . . . 6–4
6.4 Describing a multiplexer in VHDL . . . 6–5
6.4.1 Conditional Signal Assignment Statement . . . 6–5
6.4.2 Selected Signal Assignment Statement . . . 6–5
6.5 Unstructured combinational circuits . . . 6–8

7 Arithmetic combinational circuits **7–1**
7.1 Introductory concepts . . . 7–1
7.2 An Incrementer . . . 7–2
7.3 Adders . . . 7–4
7.3.1 1-bit adder . . . 7–4
7.3.2 An n-bit adder . . . 7–6
7.4 2's complement representation of numbers . . . 7–7
7.5 Changing sign of a 2's complement number . . . 7–8
7.6 Adding 2's complement numbers . . . 7–9
7.7 Carry propagation and generation . . . 7–11
7.8 Carry Lookahead adder . . . 7–12
7.9 Subtractors . . . 7–13
7.10 VHDL specification of a 1-bit adder . . . 7–14
7.11 Arithmetic-Logic Units . . . 7–16

7.12 VHDL implementation of n-bit arithmetic circuits. The "generate" statement. . . . 7–17
7.13 Structural specification of digital circuits . . . 7–18

8 Design Example: A Division-by-Constant Combinational Circuit 8–1
8.1 A general case . . . 8–1
8.2 Binary-to-decimal conversion . . . 8–2
8.3 A 1-bit division-by-constant circuit . . . 8–3
8.4 An architecture with the truth table specification . . . 8–5
8.5 An n-bit division-by-three circuit . . . 8–7

9 Sequential Circuits 9–1
9.1 Finite State Machines . . . 9–2
9.1.1 Asynchronous state machines/sequential circuits . . . 9–4
9.1.2 Synchronous (clocked) state machines . . . 9–5
9.2 Latches . . . 9–6
9.2.1 S-R Latch . . . 9–6
9.2.2 D Latch . . . 9–9
9.2.3 Latches — Summary . . . 9–11
9.2.4 VHDL description . . . 9–13
9.2.5 Simulation waveforms . . . 9–14
9.3 Flip-Flops . . . 9–15
9.4 D Flip-Flop . . . 9–16
9.4.1 VHDL description of a D flip-flop . . . 9–18
9.4.2 Logic synthesis of a D flip-flop . . . 9–19
9.4.3 Yet another implementation of a D flip-flop . . . 9–25
9.5 The T flip-flop . . . 9–26
9.5.1 Logic synthesis of the T flip-flop . . . 9–27
9.5.2 Implementing the T flip-flop with a D flip-flop . . . 9–28
9.5.3 The T flip-flop as a frequency divider . . . 9–29
9.6 The JK flip-flop . . . 9–30
9.7 Hazards in combinational and sequential circuits . . . 9–32

10 Registers and counters 10–1
10.1 Registers . . . 10–1
10.1.1 An n-bit "parallel load" register . . . 10–2
10.1.2 A simple shift register . . . 10–4
10.1.3 A bi-directional shift register . . . 10–5
10.2 Counters . . . 10–10
10.2.1 A ripple counter . . . 10–10
10.2.2 Synchronous counter . . . 10–11
10.2.3 Universal up-down counter . . . 10–12

11 Synchronous State Machines 11–1
11.1 Example of a Moore state machine . . . 11–2
11.2 Example of a Mealy state machine . . . 11–7

12 Simple serial arithmetic processor 12–1
12.1 Example: A bit-serial adder . . . 12–3

13 Multipliers 13–1
13.1 Word-Serial Multiplication Processor – the Booth's algorithm . . . 13–2
13.2 The top-level structure of the processor . . . 13–4
13.3 Datapath of the word-serial multiplication processor implementing the 1-bit Booth's algorithm . . . 13–5
13.4 State diagram of the control unit . . . 13–6
13.5 Numerical example . . . 13–7
13.6 Operations of the datapath blocks . . . 13–8
13.7 Designing the datapath . . . 13–10
13.8 The control unit . . . 13–11
13.8.1 The VHDL program for the control unit . . . 13–12
13.9 The complete word-serial multiplication processor . . . 13–16
13.10 Appendix: Multiplication methods . . . 13–17
13.11 The Booth's multiplier . . . 13–22