

---

# **STRUCTURE OF COMPUTER SYSTEMS WITH APPLICATIONS**

---



**Baruch Zoltan Francisc**

Technical University of Cluj-Napoca

Computer Science Department

**STRUCTURE OF COMPUTER  
SYSTEMS  
WITH APPLICATIONS**

U. T. PRES

Cluj-Napoca, 2003



---

# CONTENTS

---

<b>PREFACE</b> .....	<b>xi</b>
<b>1. DIDACTIC COMPUTER</b> .....	<b>1</b>
1.1. OVERVIEW OF THE DIDACTIC COMPUTER.....	1
1.2. INSTRUCTIONS OF THE DIDACTIC COMPUTER.....	2
1.2.1. Word Structure.....	2
1.2.2. Instruction Set.....	3
1.2.3. Example Programs .....	5
1.3. STRUCTURE OF THE DIDACTIC COMPUTER .....	6
1.3.1. General Structure.....	6
1.3.2. Control Signals.....	8
1.3.3. Detailed Structure.....	8
1.4. APPLICATIONS.....	10
<b>2. BINARY MULTIPLICATION AND DIVISION CIRCUITS</b> .....	<b>11</b>
2.1. BINARY MULTIPLICATION .....	11
2.1.1. Binary Multiplication Methods.....	11
2.1.2. Direct Multiplication.....	12
2.1.3. Booth Multiplication.....	18
2.1.4. Multiplication on Groups of Bits.....	23
2.2. BINARY DIVISION .....	25
2.2.1. Principle of Binary Division.....	25
2.2.2. Binary Division Methods.....	28
2.2.3. Restoring Division.....	28
2.2.4. Nonrestoring Division.....	33
2.3. APPLICATIONS.....	36
<b>3. DECIMAL MULTIPLICATION AND DIVISION CIRCUITS</b> .....	<b>38</b>
3.1. DECIMAL MULTIPLICATION .....	38
3.1.1. The Repeated-Addition Method .....	39
3.1.2. The Nine-Multiples-of-Multiplicand Method.....	39

3.1.3. The Right-and-Left-Hand Components Method.....	42
3.2. DECIMAL DIVISION.....	45
3.2.1. Restoring Division.....	46
3.2.2. Nonrestoring Division.....	47
3.2.3. The Nine-Multiples-of-Divisor Method.....	49
3.3. APPLICATIONS.....	51
<b>4. MICROPROGRAMMED COMPUTERS.....</b>	<b>52</b>
4.1. PRINCIPLE OF MICROPROGRAMMED COMPUTERS.....	52
4.1.1. Introduction.....	52
4.1.2. Horizontal and Vertical Microprogramming.....	53
4.1.3. Structure of a Microprogrammed Control Unit.....	55
4.2. EXAMPLE OF COMPUTER WITH HORIZONTAL MICROPROGRAMMING.....	57
4.2.1. Datapath Structure.....	57
4.2.2. Microinstruction Format.....	60
4.2.3. Microinstruction Sequencing.....	61
4.2.4. The Microassembly Language.....	64
4.3. EXAMPLE OF COMPUTER WITH VERTICAL MICROPROGRAMMING.....	65
4.3.1. Vertical Microinstructions.....	65
4.3.2. Structure of the Control Section.....	66
4.4. APPLICATIONS.....	69
<b>5. BASIC ELEMENTS OF THE VHDL LANGUAGE.....</b>	<b>70</b>
5.1. DESIGN UNITS.....	70
5.2. ENTITIES.....	71
5.2.1. Entity Names: Identifiers.....	71
5.2.2. Entity Declarations.....	72
5.2.3. Signals and Ports.....	73
5.2.3.1. The Concept of a Signal.....	73
5.2.3.2. Port Declarations.....	75
5.2.3.3. Port Modes.....	75
5.2.3.4. Port Types.....	77
5.2.4. Generics.....	78
5.3. ARCHITECTURES.....	79
5.3.1. Architecture Declaration.....	79
5.3.2. Signal Declaration.....	80
5.3.3. Signal Assignments.....	81
5.3.4. Styles of Architectural Descriptions.....	83
5.3.4.1. Functional Descriptions.....	83
5.3.4.2. Dataflow Descriptions.....	84
5.3.4.3. Structural Descriptions.....	86
5.3.4.4. Comparing Architectural Descriptions.....	87
5.4. MODELING FOR SIMULATION.....	89
5.4.1. Event-Driven Simulation.....	90
5.4.2. Signal Drivers.....	91

---

5.4.3. Simulation Cycle .....	94
5.4.4. Multiple Drivers and Resolution Functions .....	98
5.5. MODELING FOR SYNTHESIS .....	99
5.6. APPLICATIONS .....	101
<b>6. SEQUENTIAL AND CONCURRENT STATEMENTS IN THE VHDL LANGUAGE .....</b>	<b>104</b>
6.1. SEQUENTIAL STATEMENTS .....	104
6.1.1. Processes .....	105
6.1.1.1. Structure and Execution of a Process .....	105
6.1.1.2. Processes with Incomplete Sensitivity Lists .....	106
6.1.1.3. <code>wait</code> Statement .....	107
6.1.1.4. Combinational and Sequential Processes .....	110
6.1.2. Sequential Signal Assignment Statement .....	111
6.1.2.1. Sequential Assignment Statement Execution .....	111
6.1.2.2. Feedback .....	112
6.1.2.3. Inertial Delay .....	113
6.1.2.4. Transport Delay .....	114
6.1.3. Variables .....	115
6.1.3.1. Declaring and Initializing Variables .....	115
6.1.3.2. Variable Assignment Statement .....	116
6.1.4. <code>if</code> Statement .....	117
6.1.4.1. Syntax and Execution of an <code>if</code> Statement .....	117
6.1.4.2. Synthesis Interpretation of an <code>if</code> Statement .....	118
6.1.4.3. Incomplete <code>if</code> Statements .....	120
6.1.4.4. <code>if</code> Statements with Variables .....	124
6.1.5. <code>case</code> Statement .....	125
6.1.6. Loop Statements .....	127
6.1.6.1. <code>loop</code> Statement .....	127
6.1.6.2. <code>while loop</code> Statement .....	128
6.1.6.3. <code>for loop</code> Statement .....	129
6.1.6.4. <code>next</code> Statement .....	133
6.1.6.5. <code>exit</code> Statement .....	134
6.1.7. Sequential <code>assert</code> Statement .....	136
6.2. CONCURRENT STATEMENTS .....	137
6.2.1. Structure and Execution of an Architecture .....	138
6.2.2. Processes .....	140
6.2.3. Concurrent Signal Assignment Statements .....	140
6.2.3.1. Simple Signal Assignment .....	140
6.2.3.2. Conditional Signal Assignment .....	142
6.2.3.3. Selected Signal Assignment .....	145
6.2.3.4. <code>block</code> Statement .....	146
6.2.3.5. Concurrent <code>assert</code> Statement .....	149
6.3. EXAMPLES OF COMBINATIONAL CIRCUITS .....	149
6.3.1. Multiplexers .....	149

6.3.2. Priority Encoders.....	151
6.4. EXAMPLES OF SEQUENTIAL CIRCUITS .....	152
6.4.1. Synchronous and Asynchronous Sequential Circuits .....	152
6.4.2. Flip-Flops.....	153
6.4.3. Registers .....	155
6.4.4. Counters.....	155
6.4.5. Resetting Synchronous Logic .....	157
6.4.6. Three-State Buffers and Bidirectional Signals.....	160
6.5. APPLICATIONS .....	162
<b>7. STATE MACHINES IN THE VHDL LANGUAGE.....</b>	<b>168</b>
7.1. DESIGN EXAMPLE .....	168
7.1.1. Traditional Design.....	169
7.1.2. Design Using the VHDL Language .....	170
7.2. DESIGN OF A MEMORY CONTROLLER .....	172
7.3. TECHNIQUES TO GENERATE THE OUTPUT SIGNALS.....	179
7.3.1. Outputs Decoded from State Bits .....	179
7.3.2. Outputs Decoded in Parallel Output Registers .....	180
7.3.3. Outputs Encoded within State Bits .....	183
7.3.4. One-Hot Encoding .....	186
7.4. MEALY STATE MACHINES.....	190
7.5. OTHER DESIGN CONSIDERATIONS .....	191
7.5.1. State Encoding Using Enumeration Types .....	191
7.5.2. Explicit State Encoding .....	192
7.5.3. Fault Tolerance for One-Hot Machines .....	193
7.6. APPLICATIONS .....	194
<b>8. STRUCTURAL DESIGN IN THE VHDL LANGUAGE .....</b>	<b>197</b>
8.1. ADVANTAGES OF STRUCTURAL DESIGN .....	197
8.2. ELEMENTS OF A STRUCTURAL DESCRIPTION .....	198
8.2.1. Example of Structural Description.....	198
8.2.2. Component Declaration.....	200
8.2.3. Component Instantiation .....	201
8.2.4. Direct Entity Instantiation .....	202
8.2.5. Configuration Specification .....	203
8.3. LIBRARIES.....	206
8.4. PACKAGES .....	207
8.4.1. Package Declaration.....	207
8.4.2. Package Body .....	208
8.4.3. Packages with Component Declarations .....	209
8.5. GENERICS AND PARAMETERIZED COMPONENTS.....	210
8.5.1. Principle of Generics.....	210
8.5.2. Defining Generic Entities .....	211
8.5.3. Using Generic Components .....	212
8.5.4. Types of Generic Parameters .....	213



---

8.5.5. Building a Library of Components .....	214
8.6. GENERATE STATEMENT.....	218
8.6.1. For generate Statement .....	218
8.6.2. If generate Statement.....	220
8.6.3. Component Instances in Generate Statements.....	221
8.7. APPLICATIONS .....	223
<b>9. SUBPROGRAMS IN THE VHDL LANGUAGE.....</b>	<b>226</b>
9.1. THE ROLE OF SUBPROGRAMS .....	226
9.2. FUNCTIONS .....	227
9.2.1. Function Definition.....	227
9.2.2. Using Functions.....	230
9.2.3. Initial Values.....	231
9.2.4. Functions with Unconstrained Parameters .....	232
9.2.5. Unconstrained Return Values.....	234
9.2.6. Multiple Return Statements.....	236
9.2.7. Function Overloading.....	238
9.2.8. Type Conversion Functions.....	238
9.2.8.1. Built-In Type Conversions.....	238
9.2.8.2. User-Defined Type Conversions.....	239
9.2.9. Functions as Alternative to Component Instantiation .....	243
9.2.10. Problematic Functions for Synthesis.....	244
9.2.11. Defining Operators as Functions .....	245
9.2.11.1. Built-in Operators.....	246
9.2.11.2. Operator Overloading.....	247
9.2.12. Standard Functions .....	248
9.2.13. Placement of Function Definitions .....	250
9.3. PROCEDURES .....	252
9.3.1. Procedure Definition .....	252
9.3.2. Procedure Call.....	254
9.3.3. Procedures with Unconstrained Parameters .....	255
9.3.4. Parameters of Mode Inout.....	257
9.3.5. Signals Used as Parameters .....	257
9.3.6. Procedure Overloading.....	259
9.4. APPLICATIONS .....	261
<b>10. THE DLX COMPUTER .....</b>	<b>263</b>
10.1. ARCHITECTURE OF THE DLX COMPUTER .....	263
10.1.1. Register Set .....	263
10.1.2. Structure of the Pipelined Datapath and Instruction Execution.....	265
10.1.3. Hazards .....	266
10.2. THE DLX ASSEMBLY LANGUAGE.....	268
10.2.1. Expressions .....	268
10.2.2. Directives .....	268
10.2.3. Instruction Set.....	269

10.2.3.1. Data Transfer Instructions.....	270
10.2.3.2. Arithmetic and Logical Instructions.....	271
10.2.3.3. Control Instructions.....	273
10.2.3.4. Floating-Point Instructions.....	274
10.3. THE WINDLX SIMULATOR.....	275
10.3.1. Overview.....	275
10.3.2. System Function Calls.....	276
10.3.2.1. Open a File.....	277
10.3.2.2. Close a File.....	278
10.3.2.3. Read a Block from a File.....	278
10.3.2.4. Write a Block to a File.....	279
10.3.2.5. Formatted output to the standard output device.....	280
10.4. USING THE WINDLX SIMULATOR.....	280
10.4.1. Starting and Configuring the Simulator.....	280
10.4.2. Loading the Test Programs.....	281
10.4.3. Simulating.....	282
10.4.3.1. Pipeline Window.....	282
10.4.3.2. Code Window.....	282
10.4.3.3. Clock Cycle Diagram Window.....	283
10.4.3.4. Breakpoints Window.....	284
10.4.3.5. Register Window.....	285
10.4.3.6. Statistics Window.....	286
10.5. APPLICATIONS.....	287
<b>BIBLIOGRAPHY.....</b>	<b>290</b>