This book describes the structure of simulators suitable for use in the design of digital electronic systems. Such systems are increasing rapidly in size and complexity, and the use of simulation provides a means to ‘prototype’ a design without ever building hardware. Other advantages over hardware prototyping are that sections of a design can be simulated in isolation, and that all internal signals are available.

This book includes the compiled code and event driven algorithms for digital electronic system simulators, together with timing verification. Limitations of the structures are also discussed. An introduction to the problems of designing models is included, partly to point to how user models might be constructed for application specific integrated circuits (ASICs) and so on, and partly to expose the limitations of the modelling process.

Simulators have two functions. The first is to confirm so far as possible that a design meets its specification. The second is to check if the test program will find a sufficient percentage of possible manufacturing faults. In the former case the user must supply test vectors. In the latter, tests can be generated by automatic means. As a guide to the use of simulators the book includes chapters which introduce the subjects of testing and design for testability. A major chapter is devoted to fault simulation. Finally, the text has an introduction to hardware accelerators and modellers.

The book is suitable for electronic engineers using digital techniques, including undergraduates using design software, and postgraduates and practising engineers using simulation for the first time. It will also be useful for computer scientists needing an introduction to simulation techniques.
Electronics texts for engineers and scientists

Editors
P. L. Jones, Electrical Engineering Laboratories, University of Manchester
P. J. Spreadbury, Department of Engineering, University of Cambridge

Simulation in the design of digital electronic systems
Titles in this series

H. Ahmed and P. J. Spreadbury
*Analogue and digital electronics for engineers*

R. L. Dalglish
*An introduction to control and measurement with microcomputers*

W. Shepherd and L. N. Hulley
*Power electronics and motor control*

T. H. O’Dell
*Electronic circuit design*

K. C. A. Smith and R. E. Alley
*Electrical circuits*

A. Basak
*Analogue electronic circuits and systems*
Simulation in the design of digital electronic systems

JOHN B. GOSLING
Formerly Senior Lecturer, Department of Computer Science,
University of Manchester
Contents

Preface xiii

1 An introduction to the simulation of electronic systems 1
1.1 Introduction 1
1.2 Four aims of simulation 2
1.2.1 Functional correctness 2
1.2.2 Speed of the system 4
1.2.3 Hazard detection 5
1.2.4 Expected outputs for test and fault simulation 6
1.3 Components of a simulator 6
1.4 Levels of simulation 7
1.4.1 System design 7
1.4.2 High level 9
1.4.3 Gate level 9
1.4.4 Circuit level 10
1.4.5 Switch level 11
1.4.6 Mixed mode 12
1.5 Models 13
1.6 Test program generation 14
1.7 Fault simulation 17
1.8 Timing verification 18
1.9 Conclusion 18

2 Electronic computer aided design (ECAD) systems 20
2.1 The design process 20
2.1.1 Specification 20
viii Contents

2.1.2 Partitioning the design 22
2.1.3 Test strategy 22
2.1.4 Constructional issues 23
2.1.5 Logical design 23
2.1.6 ASIC design 24
2.1.7 Interfaces and pin limitation 25
2.2 Design capture 26
2.3 Simulation 28
2.4 Test program generation 29
2.5 Placement and routing 31
2.6 Wiring delays 34
2.7 Silicon compilation 35
2.8 Conclusion 36

3 Design for testability 37
3.1 Cost of testing 37
3.1.1 Advantages and penalties 37
3.1.2 Problem size 38
3.1.3 Combinational and sequential logic 42
3.1.4 Design for testability 43
3.2 Initialisation and resetting 43
3.3 Scan design 44
3.4 Boundary scan 47
3.5 Self-testing 51
3.5.1 Dedicated test logic 51
3.5.2 Signature analysis 51
3.5.3 Built-in logic block observation (BILBO) 52
3.6 Controllability and observability 54
3.6.1 Concepts 54
3.6.2 Controllability 56
3.6.3 Observability 60
3.6.4 Testability 61

4 Exercising the design in simulation and test 64
4.1 Objectives and approaches 64
4.1.1 Objective 64
4.1.2 Modelling faults 64
4.1.3 Assessment of test coverage 67
4.2 Testing for design faults 67
4.3 Testing for manufacturing faults 71
Contents ix

4.4 The D-algorithm 73
4.4.1 Basic ideas 73
4.4.2 Primitive D-cubes of failure 76
4.4.3 Primitive D-cubes of a logic block (propagation D-cubes) 79
4.4.4 Example of use of D-cubes 82
4.4.5 Enhancements to the D-algorithm 85
4.5 Reducing the number of test vectors 88
4.6 The MOS stuck open fault 90
4.7 High level testing 93

5 Input/output of simulation and specification of models 96
5.1 Input and output of simulation 96
5.2 Simple driver 98
5.3 Simulation output 100
5.4 Operation in parallel and in sequence 102
5.5 More general modelling facilities 107
5.5.1 WAIT 107
5.5.2 The LOOP statement 108
5.5.3 CASE statement 109

6 Simulation algorithms 111
6.1 Introduction 111
6.2 Compiled code simulation 113
6.2.1 Basic procedures 113
6.2.2 Simulator structures 115
6.2.3 Detailed example 116
6.2.4 Handling feedback 118
6.2.5 Some comments 121
6.3 Event driven simulation 122
6.3.1 Introduction 122
6.3.2 Basic procedures 124
6.4 An example – a four-gate not equivalence circuit 131
6.5 Some refinements 140
6.5.1 Affected component list and memory 140
6.5.2 Time wheel overflows 142
6.5.3 Wiring delays 143
6.6 Groups of signals 144
6.6.1 Usefulness and problems 144
6.6.2 User-defined values 145
Contents

6.6.3 Group splitting 146
6.6.4 Group combination 148

7 Models and model design 150
7.1 Some simple models 150
7.2 Delays 152
7.3 Model of a buffer 153
7.3.1 Development of the algorithm 153
7.3.2 State machine representations 158
7.3.3 Model of a simple gate 162
7.4 Inertial delay 163
7.4.1 Equal rise and fall delays 163
7.4.2 Unequal rise and fall delays 164
7.5 A three-value model 167
7.6 A five-value model 169
7.7 Logical combinations and non-logical values 170
7.8 Signal strengths 171
7.9 Towards a model for a flip-flop 175
7.9.1 More complex models 175
7.9.2 The 74xx74 style flip-flop 175
7.9.3 The preset / clear (P.C) mode 177
7.9.4 The clock controlled mode 179
7.9.5 Timing errors 181
7.10 High level modelling 183
7.10.1 Behavioural models 183
7.10.2 Hierarchical models – structural 184
7.11 Wire gates 186
7.12 Hard models 187
7.12.1 Non-memory devices 187
7.12.2 Memory 190

8 Timing verification 192
8.1 Introduction 192
8.2 Computing the critical path 193
8.3 Methods of timing verification 195
8.3.1 Path enumeration 195
8.3.2 Block orientated path trace 196
8.4 Description of the network 196
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.5</td>
<td>False paths</td>
<td>198</td>
</tr>
<tr>
<td>8.6</td>
<td>Use of timing verification</td>
<td>202</td>
</tr>
<tr>
<td>8.7</td>
<td>More complex models</td>
<td>205</td>
</tr>
<tr>
<td>9.1</td>
<td>Introduction</td>
<td>207</td>
</tr>
<tr>
<td>9.2</td>
<td>Reducing the problem size</td>
<td>207</td>
</tr>
<tr>
<td>9.2.1</td>
<td>Static reduction of tests</td>
<td>209</td>
</tr>
<tr>
<td>9.2.2</td>
<td>Dynamic reduction of the Nishida cube</td>
<td>210</td>
</tr>
<tr>
<td>9.3</td>
<td>Parallel methods of fault simulation</td>
<td>211</td>
</tr>
<tr>
<td>9.3.1</td>
<td>Single fault propagation</td>
<td>212</td>
</tr>
<tr>
<td>9.3.2</td>
<td>Extension to include faults in the wiring</td>
<td>213</td>
</tr>
<tr>
<td>9.3.3</td>
<td>Assessment</td>
<td>214</td>
</tr>
<tr>
<td>9.3.4</td>
<td>Parallel pattern single fault propagation (PPSFP)</td>
<td>215</td>
</tr>
<tr>
<td>9.3.5</td>
<td>Extension to wiring faults</td>
<td>216</td>
</tr>
<tr>
<td>9.3.6</td>
<td>Evaluation</td>
<td>218</td>
</tr>
<tr>
<td>9.3.7</td>
<td>Multiple values</td>
<td>218</td>
</tr>
<tr>
<td>9.3.8</td>
<td>Parallel fault simulation</td>
<td>219</td>
</tr>
<tr>
<td>9.3.9</td>
<td>Fault dropping</td>
<td>219</td>
</tr>
<tr>
<td>9.4</td>
<td>Concurrent fault simulation</td>
<td>220</td>
</tr>
<tr>
<td>9.4.1</td>
<td>General description</td>
<td>221</td>
</tr>
<tr>
<td>9.4.2</td>
<td>Detailed example</td>
<td>222</td>
</tr>
<tr>
<td>9.4.3</td>
<td>Change of input</td>
<td>223</td>
</tr>
<tr>
<td>9.5</td>
<td>Parallel value list (PVL)</td>
<td>224</td>
</tr>
<tr>
<td>9.5.1</td>
<td>Detailed example</td>
<td>225</td>
</tr>
<tr>
<td>9.5.2</td>
<td>Change of input</td>
<td>226</td>
</tr>
<tr>
<td>9.5.3</td>
<td>Comment</td>
<td>227</td>
</tr>
<tr>
<td>9.6</td>
<td>Assessment of simulation methods</td>
<td>228</td>
</tr>
<tr>
<td>9.6.1</td>
<td>Parallel methods of fault simulation</td>
<td>229</td>
</tr>
<tr>
<td>9.6.2</td>
<td>Concurrent fault simulation</td>
<td>230</td>
</tr>
<tr>
<td>9.6.3</td>
<td>Deductive fault simulation</td>
<td>231</td>
</tr>
<tr>
<td>9.7</td>
<td>Some alternatives to fault simulation</td>
<td>232</td>
</tr>
<tr>
<td>9.7.1</td>
<td>Critical path tracing</td>
<td>233</td>
</tr>
<tr>
<td>9.7.2</td>
<td>Statistical methods</td>
<td>234</td>
</tr>
<tr>
<td>9.7.3</td>
<td>Block orientated fault simulation</td>
<td>235</td>
</tr>
<tr>
<td>9.8</td>
<td>Timing in fault simulation</td>
<td>236</td>
</tr>
<tr>
<td>9.8.1</td>
<td>Delay faults</td>
<td>237</td>
</tr>
<tr>
<td>9.8.2</td>
<td>Oscillations and hyperactivity</td>
<td>238</td>
</tr>
</tbody>
</table>
Contents

10 Simulator features and extensions 243
  10.1 Desirable features of a simulator 243
  10.2Getting value from the simulator 246
  10.2.1 Computer aided design 246
  10.2.2 Models 247
  10.2.3 Testing functionality 248
  10.3 Wires in high speed logic 250
  10.4 Simulation accelerators 251
  10.4.1 Point acceleration 251
  10.4.2 Zycad engines 252
  10.4.3 Dazix Gigalogician 253
  10.4.4 IBM machines 254
  10.4.5 Assessment 255
  10.5 Whither now? 255

Appendix 258
References 260
Index 270
Preface

In attempting to come to grips with the problem of designing a simulator the author found very little in the way of overall descriptions of what a simulator is, what it does or how it works. The required information can be winkleed out from many different sources, but not all are easily available. This book is an attempt to bring together in one place a comprehensive introduction to all aspects of simulation in the design of digital electronic systems.

The text begins with an introduction to the purpose of simulation, types of simulation and some of the problems that are encountered in the use and design of simulators. It continues with a brief review of computer aided design suites in order to set simulation within its overall context.

In order to use a simulator it is necessary to prepare test information. To get the best out of the simulator it is necessary to adopt good design techniques. Hence the next two chapters give an introduction to design for testability and to test program generation. These are followed by a brief description of the preparation of test programs using the VHPIC high level design language (VHDL). These three chapters are just an introduction for completeness in the book as a whole, and the reader is referred to much more comprehensive texts for a proper treatment.

Chapters 6 to 9 are the meat of this work. Chapter 6 describes the two main types of straightforward simulator and gives some examples of their use. That is followed by a description of a method by which the necessary models can be written. Chapter 8 deals with timing verification and Chapter 9 with fault simulation. So far as is known this is the only introduction to modelling or to timing verification in one place. Chapter 6 is also the most comprehensive description of simulators known.
xiv  Preface

Whilst in some ways Chapters 4 and 9 go together, the topics are put in
the order as here because that is the order in which the user would meet the
techniques in a typical design exercise. That is to say, a designer must
consider design for test and testability very early, but will then do much
functional simulation before beginning the test program generation and
fault simulation in earnest.

Although mentioned in Chapter 1, circuit simulation and switch level
modelling are not described in detail. Circuit simulation uses very different
techniques to digital work and is beyond the scope of this book. Much
switch level simulation can be handled by the techniques described in
Chapters 6, 8 and 9, but some additional modelling considerations become
important. Some hints to these are given but a full description is omitted for
brevity.

The intended readership is undergraduates in Electrical and Computer
Engineering and those studying computer aided design of electronic
systems in Computer Science. It is aimed at both users of simulators and at
those who may wish eventually to be involved in their design. Whilst some
sections of the book will be more important to one group and other parts to
a different group, the whole book is relevant to all. An understanding of the
working of the simulator will lead to better use of it, and an understanding
of the needs of users will lead to better design. In particular, whilst users will
normally use models supplied with the CAD suite, they will still have to
write high level models for their own design, possibly at a fairly detailed
level.

A minimum of assumptions about the knowledge of the reader has been
made. The primary assumptions are that the reader understand the basic
operation of a gate and a flip-flop, although Chapter 1 contains a brief
description of the working of a flip-flop. It is assumed that the reader
understands the concept of there being a delay between application of an
input to a circuit and the output changing. An indication of how this can
become more complex is given. An awareness of the concepts of set-up and
hold time would be helpful but not essential. Attention is drawn to the
distinction between the terms ‘latch’ and ‘flip-flop’ given as a footnote in
Section 3.2.

Some understanding of simple computer data structures will be helpful,
as will some elementary knowledge of electronic circuits – current flow,
input and output limitations etc. In both cases readers without this
knowledge will need to take some statements on trust but should not find
the lack of background serious. Diagrams of logical devices more complex
than a gate are very few and use the dependency notation. Readers
unfamiliar with this should investigate it as it makes the function of
Preface

modules very easy to understand.

Conventional drawing conventions are assumed. That is, signal flow is left to right and top to bottom unless otherwise indicated. Logic is drawn as nearly as possible to the IEEE standard on dependency notation. Digital signals are usually written as 1 or 0 as appropriate to indicate ‘active’ and ‘inactive’ respectively. Signal values are written in italic font to distinguish from numbers or literal signal names.

Some of the references are used in the text, but others are added as pointers to additional reading. In many cases some comments are included. The selection of references is inevitably that of the author. It is limited to those which describe basic ideas well or which, in the author’s opinion, are most likely to be of long term use. Since early papers often come into the former category there are rather more of them than might seem proper at first sight.

There are no tutorial questions provided as such. As Chapters 3 to 5 are introductory only, the reader should refer to specialist texts. In Chapters 6 to 9 some simple examples such as full adders or even just a two to one multiplexer can be used. It is not difficult to work out from knowledge of these devices what results the simulation and model should give and thus provide a check on the working of the example. In some places, notably in Chapter 9, some extensions of the example in the text are suggested and results provided.

Acknowledgements

The author owes a considerable debt to Miss Hilary J. Kahn of the University of Manchester, with whom he has worked both as a user and as a designer of simulators over many years. Dr Andrew F. Carpenter, also at Manchester, has commented on the initial text and has assisted with a number of matters, in particular with the VHDL and fault simulator work. Daniel Cock also made several very important contributions in relation to fault simulators. Peter L. Jones, the publisher’s series editor and another former colleague in Manchester has been most helpful throughout. The University of Manchester (through Dr Carpenter) has also provided facilities to check the VHDL and to check the models by use of VHDL. Lastly, the author’s wife has suffered much and long in the process of getting the text to market.

John B. Gosling
Glossop