Digital Integrated Circuit Design
From VLSI Architectures to CMOS Fabrication

Hubert Kaeslin

ETH Zürich
# Contents

## Preface

### page xix

## Acknowledgements

### xxiii

### Chapter 1 | Introduction to Microelectronics

- 1.1 Economic impact
- 1.2 Concepts and terminology
  - 1.2.1 The Guinness book of records point of view
  - 1.2.2 The marketing point of view
  - 1.2.3 The fabrication point of view
  - 1.2.4 The design engineer's point of view
  - 1.2.5 The business point of view
- 1.3 Design flow in digital VLSI
  - 1.3.1 The Y-chart, a map of digital electronic systems
  - 1.3.2 Major stages in VLSI design
  - 1.3.3 Cell libraries
  - 1.3.4 Electronic design automation software
- 1.4 Field-programmable logic
  - 1.4.1 Configuration technologies
  - 1.4.2 Organization of hardware resources
  - 1.4.3 Commercial products
- 1.5 Problems
- 1.6 Appendix I: A brief glossary of logic families
- 1.7 Appendix II: An illustrated glossary of circuit-related terms

### Chapter 2 | From Algorithms to Architectures

- 2.1 The goals of architecture design
  - 2.1.1 Agenda
- 2.2 The architectural antipodes
  - 2.2.1 What makes an algorithm suitable for a dedicated VLSI architecture?
  - 2.2.2 There is plenty of land between the architectural antipodes
  - 2.2.3 Assemblies of general-purpose and dedicated processing units
  - 2.2.4 Coprocessors
  - 2.2.5 Application-specific instruction set processors
  - 2.2.6 Configurable computing
  - 2.2.7 Extendable instruction set processors
  - 2.2.8 Digest
- 2.3 A transform approach to VLSI architecture design
  - 2.3.1 There is room for remodelling in the algorithmic domain
  - 2.3.2 ... and there is room in the architectural domain
  - 2.3.3 Systems engineers and VLSI designers must collaborate
  - 2.3.4 A graph-based formalism for describing processing algorithms
## Chapter 3 | Functional Verification

3.1 How to establish valid functional specifications
   - 3.1.1 Formal specification
   - 3.1.2 Rapid prototyping

3.2 Developing an adequate simulation strategy
   - 3.2.1 What does it take to uncover a design flaw during simulation?
   - 3.2.2 Stimulation and response checking must occur automatically
   - 3.2.3 Exhaustive verification remains an elusive goal
   - 3.2.4 All partial verification techniques have their pitfalls
   - 3.2.5 Collecting test cases from multiple sources helps
   - 3.2.6 Assertion-based verification helps
   - 3.2.7 Separating test development from circuit design helps
   - 3.2.8 Virtual prototypes help to generate expected responses

3.3 Reusing the same functional gauge throughout the entire design cycle
   - 3.3.1 Alternative ways to handle stimuli and expected responses
   - 3.3.2 Modular testbench design
   - 3.3.3 A well-defined schedule for stimuli and responses
   - 3.3.4 Trimming run times by skipping redundant simulation sequences
   - 3.3.5 Abstracting to higher-level transactions on higher-level data
   - 3.3.6 Absorbing latency variations across multiple circuit models

3.4 Conclusions

3.5 Problems

3.6 Appendix I: Formal approaches to functional verification

3.7 Appendix II: Deriving a coherent schedule for simulation and test

## Chapter 4 | Modelling Hardware with VHDL

4.1 Motivation
   - 4.1.1 Why hardware synthesis?
   - 4.1.2 What are the alternatives to VHDL?
   - 4.1.3 What are the origins and aspirations of the IEEE 1076 standard?
   - 4.1.4 Why bother learning hardware description languages?
   - 4.1.5 Agenda

4.2 Key concepts and constructs of VHDL
   - 4.2.1 Circuit hierarchy and connectivity
   - 4.2.2 Concurrent processes and process interaction
   - 4.2.3 A discrete replacement for electrical signals
   - 4.2.4 An event-based concept of time for governing simulation
   - 4.2.5 Facilities for model parametrization
   - 4.2.6 Concepts borrowed from programming languages

4.3 Putting VHDL to service for hardware synthesis
   - 4.3.1 Synthesis overview
4.3.2 Data types 224  
4.3.3 Registers, finite state machines, and other sequential subcircuits 225  
4.3.4 RAMs, ROMs, and other macrocells 231  
4.3.5 Circuits that must be controlled at the netlist level 233  
4.3.6 Timing constraints 234  
4.3.7 Limitations and caveats for synthesis 238  
4.3.8 How to establish a register transfer-level model step by step 238  
4.4 Putting VHDL to service for hardware simulation 242  
4.4.1 Ingredients of digital simulation 242  
4.4.2 Anatomy of a generic testbench 242  
4.4.3 Adapting to a design problem at hand 245  
4.4.4 The VITAL modelling standard IEEE 1076.4 245  
4.5 Conclusions 247  
4.6 Problems 248  
4.7 Appendix I: Books and Web Pages on VHDL 250  
4.8 Appendix II: Related extensions and standards 251  
4.8.1 Protected shared variables IEEE 1076a 251  
4.8.2 The analog and mixed-signal extension IEEE 1076.1 252  
4.8.3 Mathematical packages for real and complex numbers IEEE 1076.2 253  
4.8.4 The arithmetic packages IEEE 1076.3 254  
4.8.5 A language subset earmarked for synthesis IEEE 1076.6 254  
4.8.6 The standard delay format (SDF) IEEE 1497 254  
4.8.7 A handy compilation of type conversion functions 255  
4.9 Appendix III: Examples of VHDL models 256  
4.9.1 Combinational circuit models 256  
4.9.2 Mealy, Moore, and Medvedev machines 261  
4.9.3 State reduction and state encoding 268  
4.9.4 Simulation testbenches 270  
4.9.5 Working with VHDL tools from different vendors 285  

Chapter 5  The Case for Synchronous Design 286  
5.1 Introduction 286  
5.2 The grand alternatives for regulating state changes 287  
  5.2.1 Synchronous clocking 287  
  5.2.2 Asynchronous clocking 288  
  5.2.3 Self-timed clocking 288  
5.3 Why a rigorous approach to clocking is essential in VLSI 290  
  5.3.1 The perils of hazards 290  
  5.3.2 The pros and cons of synchronous clocking 291  
  5.3.3 Clock-as-clock-can is not an option in VLSI 293  
  5.3.4 Fully self-timed clocking is not normally an option either 294  
  5.3.5 Hybrid approaches to system clocking 294  
5.4 The dos and don'ts of synchronous circuit design 296  
  5.4.1 First guiding principle: Dissociate signal classes! 296  
  5.4.2 Second guiding principle: Allow circuits to settle before clocking! 298
<table>
<thead>
<tr>
<th>Chapter 6</th>
<th>Clocking of Synchronous Circuits</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 What is the difficulty in clock distribution?</td>
<td>315</td>
</tr>
<tr>
<td>6.1.1 Agenda</td>
<td>315</td>
</tr>
<tr>
<td>6.1.2 Timing quantities related to clock distribution</td>
<td>317</td>
</tr>
<tr>
<td>6.2 How much skew and jitter does a circuit tolerate?</td>
<td>317</td>
</tr>
<tr>
<td>6.2.1 Basics</td>
<td>317</td>
</tr>
<tr>
<td>6.2.2 Single-edge-triggered one-phase clocking</td>
<td>319</td>
</tr>
<tr>
<td>6.2.3 Dual-edge-triggered one-phase clocking</td>
<td>326</td>
</tr>
<tr>
<td>6.2.4 Symmetric level-sensitive two-phase clocking</td>
<td>327</td>
</tr>
<tr>
<td>6.2.5 Unsymmetric level-sensitive two-phase clocking</td>
<td>331</td>
</tr>
<tr>
<td>6.2.6 Single-wire level-sensitive two-phase clocking</td>
<td>334</td>
</tr>
<tr>
<td>6.2.7 Level-sensitive one-phase clocking and wave pipelining</td>
<td>336</td>
</tr>
<tr>
<td>6.3 How to keep clock skew within tight bounds</td>
<td>339</td>
</tr>
<tr>
<td>6.3.1 Clock waveforms</td>
<td>339</td>
</tr>
<tr>
<td>6.3.2 Collective clock buffers</td>
<td>340</td>
</tr>
<tr>
<td>6.3.3 Distributed clock buffer trees</td>
<td>343</td>
</tr>
<tr>
<td>6.3.4 Hybrid clock distribution networks</td>
<td>344</td>
</tr>
<tr>
<td>6.3.5 Clock skew analysis</td>
<td>345</td>
</tr>
<tr>
<td>6.4 How to achieve friendly input/output timing</td>
<td>346</td>
</tr>
<tr>
<td>6.4.1 Friendly as opposed to unfriendly I/O timing</td>
<td>346</td>
</tr>
<tr>
<td>6.4.2 Impact of clock distribution delay on I/O timing</td>
<td>347</td>
</tr>
<tr>
<td>6.4.3 Impact of PTV variations on I/O timing</td>
<td>349</td>
</tr>
<tr>
<td>6.4.4 Registered inputs and outputs</td>
<td>350</td>
</tr>
<tr>
<td>6.4.5 Adding artificial contamination delay to data inputs</td>
<td>350</td>
</tr>
<tr>
<td>6.4.6 Driving input registers from an early clock</td>
<td>351</td>
</tr>
<tr>
<td>6.4.7 Tapping a domain’s clock from the slowest component therein</td>
<td>351</td>
</tr>
<tr>
<td>6.4.8 “Zero-delay” clock distribution by way of a DLL or PLL</td>
<td>352</td>
</tr>
<tr>
<td>6.5 How to implement clock gating properly</td>
<td>353</td>
</tr>
<tr>
<td>6.5.1 Traditional feedback-type registers with enable</td>
<td>353</td>
</tr>
<tr>
<td>6.5.2 A crude and unsafe approach to clock gating</td>
<td>354</td>
</tr>
<tr>
<td>6.5.3 A simple clock gating scheme that may work under certain conditions</td>
<td>355</td>
</tr>
</tbody>
</table>
### Chapter 6

6.5.4 Safe clock gating schemes 355
6.6 Summary 357
6.7 Problems 361

### Chapter 7

7.1 Motivation 364
7.2 The data consistency problem of vectored acquisition 366
  7.2.1 Plain bit-parallel synchronization 366
  7.2.2 Unit-distance coding 367
  7.2.3 Suppression of crossover patterns 368
  7.2.4 Handshaking 369
  7.2.5 Partial handshaking 371
7.3 The data consistency problem of scalar acquisition 373
  7.3.1 No synchronization whatsoever 373
  7.3.2 Synchronization at multiple places 373
  7.3.3 Synchronization at a single place 373
  7.3.4 Synchronization from a slow clock 374
7.4 Metastable synchronizer behavior 374
  7.4.1 Marginal triggering and how it becomes manifest 374
  7.4.2 Repercussions on circuit functioning 378
  7.4.3 A statistical model for estimating synchronizer reliability 379
  7.4.4 Plesiochronous interfaces 381
  7.4.5 Containment of metastable behavior 381
7.5 Summary 384
7.6 Problems 384

### Chapter 8

8.1 CMOS logic gates 386
  8.1.1 The MOSFET as a switch 387
  8.1.2 The inverter 388
  8.1.3 Simple CMOS gates 396
  8.1.4 Composite or complex gates 399
  8.1.5 Gates with high-impedance capabilities 403
  8.1.6 Parity gates 406
  8.1.7 Adder slices 407
8.2 CMOS bistables 409
  8.2.1 Latches 410
  8.2.2 Function latches 412
  8.2.3 Single-edge-triggered flip-flops 413
  8.2.4 The mother of all flip-flops 415
  8.2.5 Dual-edge-triggered flip-flops 417
  8.2.6 Digest 418
8.3 CMOS on-chip memories 418
9.5.3 Adiabatic logic

Chapter 10 | Signal Integrity

10.1 Introduction
  10.1.1 How does noise enter electronic circuits?
  10.1.2 How does noise affect digital circuits?
  10.1.3 Agenda
10.2 Crosstalk
10.3 Ground bounce and supply droop
  10.3.1 Coupling mechanisms due to common series impedances
  10.3.2 Where do large switching currents originate?
  10.3.3 How severe is the impact of ground bounce?
10.4 How to mitigate ground bounce
  10.4.1 Reduce effective series impedances
  10.4.2 Separate polluters from potential victims
  10.4.3 Avoid excessive switching currents
  10.4.4 Safeguard noise margins
10.5 Conclusions
10.6 Problems
10.7 Appendix: Derivation of second-order approximation

Chapter 11 | Physical Design

11.1 Agenda
11.2 Conducting layers and their characteristics
  11.2.1 Geometric properties and layout rules
  11.2.2 Electrical properties
  11.2.3 Connecting between layers
  11.2.4 Typical roles of conducting layers
11.3 Cell-based back-end design
  11.3.1 Floorplanning
  11.3.2 Identify major building blocks and clock domains
  11.3.3 Establish a pin budget
  11.3.4 Find a relative arrangement of all major building blocks
  11.3.5 Plan power, clock, and signal distribution
  11.3.6 Place and route (P&R)
  11.3.7 Chip assembly
11.4 Packaging
  11.4.1 Wafer sorting
  11.4.2 Wafer testing
  11.4.3 Backgrinding and singulation
  11.4.4 Encapsulation
  11.4.5 Final testing and binning
  11.4.6 Bonding diagram and bonding rules
  11.4.7 Advanced packaging techniques
Chapter 13 | VLSI Economics and Project Management

13.1 Agenda
13.2 Models of industrial cooperation
  13.2.1 Systems assembled from standard parts exclusively
  13.2.2 Systems built around program-controlled processors
  13.2.3 Systems designed on the basis of field-programmable logic
  13.2.4 Systems designed on the basis of semi-custom ASICs
  13.2.5 Systems designed on the basis of full-custom ASICs
13.3 Interfacing within the ASIC industry
  13.3.1 Handoff points for IC design data
  13.3.2 Scopes of IC manufacturing services
13.4 Virtual components
  13.4.1 Copyright protection vs. customer information
  13.4.2 Design reuse demands better quality and more thorough verification
  13.4.3 Many existing virtual components need to be reworked
  13.4.4 Virtual components require follow-up services
  13.4.5 Indemnification provisions
  13.4.6 Deliverables of a comprehensive VC package
  13.4.7 Business models
13.5 The costs of integrated circuits
  13.5.1 The impact of circuit size
  13.5.2 The impact of the fabrication process
  13.5.3 The impact of volume
  13.5.4 The impact of configurability
  13.5.5 Digest
13.6 Fabrication avenues for small quantities
  13.6.1 Multi-project wafers
  13.6.2 Multi-layer reticles
  13.6.3 Electron beam lithography
  13.6.4 Laser programming
  13.6.5 Hardwired FPGAs and structured ASICs
  13.6.6 Cost trading
13.7 The market side
  13.7.1 Ingredients of commercial success
  13.7.2 Commercialization stages and market priorities
  13.7.3 Service versus product
  13.7.4 Product grading
13.8 Making a choice
  13.8.1 ASICs yes or no?
  13.8.2 Which implementation technique should one adopt?
  13.8.3 What if nothing is known for sure?
  13.8.4 Can system houses afford to ignore microelectronics?
13.9 Keys to successful VLSI design
  13.9.1 Project definition and marketing
Chapter 14 | A Primer on CMOS Technology

14.1 The essence of MOS device physics
  14.1.1 Energy bands and electrical conduction
  14.1.2 Doping of semiconductor materials
  14.1.3 Junctions, contacts, and diodes
  14.1.4 MOSFETs

14.2 Basic CMOS fabrication flow
  14.2.1 Key characteristics of CMOS technology
  14.2.2 Front-end-of-line fabrication steps
  14.2.3 Back-end-of-line fabrication steps
  14.2.4 Process monitoring
  14.2.5 Photolithography

14.3 Variations on the theme
  14.3.1 Copper has replaced aluminum as interconnect material
  14.3.2 Low-permittivity interlevel dielectrics are replacing silicon dioxide
  14.3.3 High-permittivity gate dielectrics to replace silicon dioxide
  14.3.4 Strained silicon and SiGe technology
  14.3.5 Metal gates bound to come back
  14.3.6 Silicon-on-insulator (SOI) technology

Chapter 15 | Outlook

15.1 Evolution paths for CMOS technology
  15.1.1 Classic device scaling
  15.1.2 The search for new device topologies
  15.1.3 Vertical integration
  15.1.4 The search for better semiconductor materials

15.2 Is there life after CMOS?
  15.2.1 Non-CMOS data storage
  15.2.2 Non-CMOS data processing

15.3 Technology push
  15.3.1 The so-called industry “laws” and the forces behind them
  15.3.2 Industrial roadmaps

15.4 Market pull
15.5 Evolution paths for design methodology 724
   15.5.1 The productivity problem 724
   15.5.2 Fresh approaches to architecture design 727
15.6 Summary 729
15.7 Six grand challenges 730
15.8 Appendix: Non-semiconductor storage technologies for comparison 731

Appendix A | Elementary Digital Electronics 732

A.1 Introduction 732
   A.1.1 Common number representation schemes 732
   A.1.2 Notational conventions for two-valued logic 734
A.2 Theoretical background of combinational logic 735
   A.2.1 Truth table 735
   A.2.2 The n-cube 736
   A.2.3 Karnaugh map 736
   A.2.4 Program code and other formal languages 736
   A.2.5 Logic equations 737
   A.2.6 Two-level logic 738
   A.2.7 Multilevel logic 740
   A.2.8 Symmetric and monotone functions 741
   A.2.9 Threshold functions 741
   A.2.10 Complete gate sets 742
   A.2.11 Multi-output functions 742
   A.2.12 Logic minimization 743
A.3 Circuit alternatives for implementing combinational logic 747
   A.3.1 Random logic 747
   A.3.2 Programmable logic array (PLA) 747
   A.3.3 Read-only memory (ROM) 749
   A.3.4 Array multiplier 749
   A.3.5 Digest 750
A.4 Bistables and other memory circuits 751
   A.4.1 Flip-flops or edge-triggered bistables 752
   A.4.2 Latches or level-sensitive bistables 755
   A.4.3 Unclocked bistables 756
   A.4.4 Random access memories (RAMs) 760
A.5 Transient behavior of logic circuits 761
   A.5.1 Glitches, a phenomenological perspective 762
   A.5.2 Function hazards, a circuit-independent mechanism 763
   A.5.3 Logic hazards, a circuit-dependent mechanism 764
   A.5.4 Digest 765
A.6 Timing quantities 766
   A.6.1 Delay quantities apply to combinational and sequential circuits 766
   A.6.2 Timing conditions apply to sequential circuits only 768
   A.6.3 Secondary timing quantities 770
   A.6.4 Timing constraints address synthesis needs 771
## CONTENTS

A.7 Microprocessor input/output transfer protocols 771
A.8 Summary 773

### Appendix B  Finite State Machines 775

B.1 Abstract automata 775
  B.1.1 Mealy machine 776
  B.1.2 Moore machine 777
  B.1.3 Medvedev machine 778
  B.1.4 Relationships between finite state machine models 779
  B.1.5 Taxonomy of finite state machines 782
  B.1.6 State reduction 783

B.2 Practical aspects and implementation issues 785
  B.2.1 Parasitic states and symbols 785
  B.2.2 Mealy-, Moore-, Medvedev-type, and combinational output bits 787
  B.2.3 Through paths and logic instability 787
  B.2.4 Switching hazards 789
  B.2.5 Hardware costs 790

B.3 Summary 793

### Appendix C  VLSI Designer’s Checklist 794

C.1 Design data sanity 794
C.2 Pre-synthesis design verification 794
C.3 Clocking 795
C.4 Gate-level considerations 796
C.5 Design for test 797
C.6 Electrical considerations 798
C.7 Pre-layout design verification 799
C.8 Physical considerations 800
C.9 Post-layout design verification 800
C.10 Preparation for testing of fabricated prototypes 801
C.11 Thermal considerations 802
C.12 Board-level operation and testing 802
C.13 Documentation 802

### Appendix D  Symbols and constants 804

D.1 Mathematical symbols used 804
D.2 Abbreviations 807
D.3 Physical and material constants 808

References 811
Index 832