

1. Record Nr.	UNINA9910627238803321
Autore	Taraate Vaibbhav
Titolo	Digital design from the VLSI perspective : concepts for VLSI beginners / / Vaibbhav Taraate
Pubbl/distr/stampa	Singapore : , : Springer, , [2023] ©2023
ISBN	9789811946523 9789811946516
Descrizione fisica	1 online resource (309 pages)
Disciplina	621.3815
Soggetti	Digital integrated circuits - Design and construction Integrated circuits - Very large scale integration Digital integrated circuits
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Intro -- Preface -- Acknowledgements -- Contents -- About the Author -- 1 Introduction -- 1.1 Number Representation -- 1.2 Digital Systems: System Perspective -- 1.3 Processors and Their Role -- 1.4 The Important Terminology: System Perspective -- 1.5 System Design Components -- 1.6 Few Important Considerations -- 1.7 Summary -- 2 Basics of Design Elements -- 2.1 Combinational Design Elements -- 2.1.1 Logic Gates and Their Use in the Design -- 2.2 De Morgan's Theorems -- 2.2.1 NAND is Equal to Bubbled OR -- 2.2.2 NOR is Equal to Bubbled and -- 2.3 Level Versus Edge Sensitive Elements -- 2.3.1 Latches and Their Use in the Design -- 2.3.2 Edge Sensitive Elements and Their Role -- 2.4 Summary -- 3 System and Architecture Design -- 3.1 Architecture of the Design -- 3.2 Micro-Architecture of the Design -- 3.3 System Design Architecture -- 3.4 Design for the Glue Logic -- 3.5 Application of 2-variable Karnaugh Maps -- 3.6 Let Us Design Two Variable Function -- 3.7 SOP Terms and Boolean Expression -- 3.8 POS Terms and Expression -- 3.9 Design of Glue or Combinational Using Minimum Logic Gates -- 3.10 Summary -- 4 Combinational Logic and Design Techniques -- 4.1 Let Us Design Few Boolean Functions -- 4.2 Arithmetic Resources -- 4.2.1

Half Adder -- 4.2.2 Half Subtractor -- 4.2.3 Full Adder -- 4.2.4 Full Adder Using Half Adders -- 4.3 Role of Data Control Elements -- 4.4 The Multi-Bit Adder and Subtractor -- 4.5 The Multi-Bit Adder with Area Optimization -- 4.6 3-Variable K-Map and Code Converters -- 4.6.1 3-bit Binary to Gray Code Converter -- 4.6.2 3-Bit Gray to Binary Code Converter -- 4.7 Summary -- 5 Data Control Elements and Applications -- 5.1 Data and Control Paths in Design -- 5.2 Multiplexers to Control the Data -- 5.3 Lowest Order Mux in the Design -- 5.4 The 2:1 MUX Using NAND -- 5.5 The 4:1 MUX -- 5.6 The Design of 4:1 Mux Using 2:1 Mux. -- 5.7 Design Using Multiplexers -- 5.7.1 NOT Using 2:1 Mux -- 5.7.2 NAND Using Mux -- 5.7.3 NOR Using 2:1 Mux -- 5.7.4 Design of XOR Gate to Get the SUM Output Using Mux -- 5.7.5 Design of XNOR Gate to Get the Even Parity -- 5.8 Boolean Functions and Implementation Using Mux -- 5.9 Mux as Universal Logic -- 5.10 Summary -- 6 Decoders and Encoders -- 6.1 Demultiplexers and Use in the Design -- 6.2 Decoder 2 to 4 Having Active High Output -- 6.3 Decoder 2 to 4 Having Active Low Output -- 6.4 Design for the Given Specifications -- 6.5 Design of 3:8 Encoder Using 2:4 Decoders -- 6.6 Encoders and Their Applications -- 6.7 Practical Encoder Design -- 6.8 Priority Encoders -- 6.9 Practical Design Scenario -- 6.10 Summary -- 7 Combinational Design Scenarios -- 7.1 Mux-Based Designs and Optimization -- 7.2 Right and Left Shift Using Multiplexers -- 7.3 Design of 8:1 Mux Using 4:1 Mux -- 7.4 Design of 8:1 Mux Using 2:1 Mux -- 7.5 Boolean Expression from the Logic -- 7.6 Boolean Expression for Mux-Based Design -- 7.7 Stuck at Faults -- 7.8 Design Using Decoders -- 7.9 Design Using Decoder and NAND Gates -- 7.10 Summary -- 8 Synchronous Sequential Design -- 8.1 Sequential Design Elements -- 8.2 Synchronous Design -- 8.3 Why to Use Synchronous Design? -- 8.4 Asynchronous Design -- 8.4.1 D Flip-Flop and Use in the Design -- 8.5 Design of the Synchronous Counters -- 8.6 Design of the Synchronous Down-Counters -- 8.7 Design of the Synchronous Gray Counter -- 8.8 Few Important Guidelines -- 8.9 Summary -- 9 Logic Design Scenarios and Objectives -- 9.1 What is Asynchronous Design? -- 9.2 Synchronous Versus Asynchronous Reset -- 9.2.1 D Flip-Flop Having Asynchronous Reset -- 9.2.2 Synchronous Reset D Flip\_flop -- 9.3 Asynchronous MOD Counters -- 9.3.1 Frequency Divider Network -- 9.3.2 Ripple Counter Design -- 9.4 Design Scenario -- 9.5 PIPO Register. -- 9.6 Shift Register -- 9.6.1 Shift Operation and Clock Cycles -- 9.7 Bidirectional Shift Register -- 9.8 Important Design Guidelines -- 9.9 Summary -- 10 Sequential Design Scenarios -- 10.1 Design Scenario I -- 10.2 Four-Bit Latch -- 10.3 Positive Edge Sensitive Flip-Flop Using Multiplexers -- 10.4 Flip-Flop Negative Edge Sensitive -- 10.5 Timing Sequence of Design -- 10.6 Load and Shift Register -- 10.7 Design Scenario II -- 10.8 Design Scenario III -- 10.9 Design Scenario III -- 10.10 Design Scenario IV -- 10.11 Design of 4-bit Ring Counter -- 10.12 Design of 4-bit Johnson Counter -- 10.13 Duty Cycle Control -- 10.13.1 Counter Design with 50% Duty Cycle -- 10.14 Summary -- 11 Timing Parameters and Maximum Frequency Calculations -- 11.1 What is Delay in the System? -- 11.1.1 Cascade Logic Elements in Design -- 11.1.2 Parallel Logic Elements in Design -- 11.2 How Delays Affect the Performance of the Design? -- 11.3 Sequential Circuit and Timing Parameters -- 11.4 Timing Paths in Design -- 11.4.1 Input to Register Path -- 11.4.2 Register to Output Path -- 11.4.3 Register to Register Path -- 11.4.4 Input to Output Path -- 11.5 Maximum Frequency Calculations -- 11.5.1 Design 1: Toggle Flip-Flop -- 11.5.2 Design II: The 2-bit Synchronous Up-Counter -- 11.6 Maximum Operating

Frequency -- 11.6.1 Maximum Operating Frequency for Synchronous Designs -- 11.7 Clock Skew -- 11.7.1 Positive Clock Skew and Maximum Operating Frequency -- 11.7.2 Negative Clock Skew and Maximum Operating Frequency for the Design -- 11.8 VLSI Specific Scenarios -- 11.8.1 VLSI Specific Design Scenario I -- 11.8.2 VLSI Specific Design Scenario II -- 11.9 Hold Slack -- 11.9.1 VLSI Specific Design Scenario III -- 11.10 Summary -- 12 FSM Designs -- 12.1 Introduction to FSM -- 12.1.1 Moore FSM -- 12.1.2 Mealy FSM -- 12.1.3 Moore Versus Mealy FSM -- 12.2 State Encoding Methods. 12.3 Moore FSM Design -- 12.4 Mealy FSM Design -- 12.5 Applications and Design Strategies -- 12.6 State Diagrams -- 12.6.1 Moore Machine State Diagram -- 12.6.2 Mealy Machine State Diagram -- 12.7 Summary -- 13 Design of Sequence Detectors -- 13.1 Moore Machine Non-overlapping 101 Sequence Detector -- 13.2 Mealy Machine Non-overlapping 101 Sequence Detector -- 13.3 One-Hot Encoding -- 13.4 FSM Area and Power Optimization -- 13.5 Moore Sequence Detector for 101 Overlapping Sequence -- 13.6 Mealy Sequence Detector for 101 Overlapping Sequence -- 13.7 Mealy Sequence Detector for 1010 Overlapping Sequence -- 13.8 Various Paths in the Design -- 13.9 Data and Control Path Design Techniques -- 13.10 Summary -- 14 Performance Improvement for the Design -- 14.1 What Is Design Performance? -- 14.2 How to Use the Minimum Arithmetic Resources -- 14.3 Multibit Adders and Subtractors -- 14.4 Four-Bit Full Adder -- 14.4.1 4-Bit Full Subtractor -- 14.4.2 4-Bit Adder and Subtractor -- 14.4.3 Area Optimization of 4-Bit Adder and Subtractor -- 14.4.4 Optimization of Design Using Only Adders -- 14.4.5 Optimization by Tweaking the Logic to Have Least Area and Least Power -- 14.5 Optimization of the Design for Least Area and Power -- 14.6 Comparators and Parity Detectors with Lesser Area -- 14.6.1 Binary Comparator Design with Least Area -- 14.6.2 Parity Detector Design with Least Area -- 14.7 Processor Designs and Speed Improvement Techniques (Source: [www.onerupeest.com](http://www.onerupeest.com)) -- 14.8 Avoid Asynchronous Designs to Improve the Speed -- 14.9 Power Improvement -- 14.9.1 Gated Clocks and Dynamic Power Reduction -- 14.10 Summary -- 15 Optimization Techniques -- 15.1 Let Us Understand About the Area Optimization -- 15.2 Arithmetic Resource Sharing -- 15.3 Resource Sharing for Sequential Circuits -- 15.4 Logic Duplications -- 15.5 Design Scenario: Performance Improvement. 15.6 Use of Pipelining in Design -- 15.6.1 Design Without Pipelining -- 15.6.2 Speed Improvement Using Register Balancing or Pipelining -- 15.7 Power Improvement of Design -- 15.8 Dynamic Power Reduction -- 15.9 Summary -- 16 Case Study: Speed Improvement for the Design -- 16.1 Case Study: Speed Improvement at Logic-Level Case Study -- 16.2 Speed Improvement at Architecture Level (Source: [www.onerupeest.com](http://www.onerupeest.com)) -- 16.2.1 Top-Level Pin Interface -- 16.2.2 Pin Description -- 16.2.3 Case Study: Micro-architecture Design -- 16.3 Summary -- 17 Case Study: Multiple Clock Domains and FIFO Architecture Design -- 17.1 Single Clock Domain Designs -- 17.2 Multiple Clock Domain Designs -- 17.3 Metastability -- 17.4 Control Path Synchronizer -- 17.5 Data Path Synchronizers -- 17.5.1 Why We Need FIFO? -- 17.5.2 FIFO Depth Calculation -- 17.5.3 Case Study: FIFO as a Data Path Synchronizer -- 17.5.4 Micro-architecture of FIFO -- 17.6 Design Guidelines -- 17.7 Summary -- 18 Hardware Description for Design -- 18.1 Verilog HDL -- 18.2 Use of Continuous Assignments -- 18.3 The always Procedural Block -- 18.4 The Procedural Block always@\* -- 18.5 Use of the case Construct -- 18.6 Continuous Versus Procedural Assignments -- 18.7 Multiple Blocking Assignments Within the always Block -- 18.8 Design Scenario I:

Blocking Assignments -- 18.9 Non-blocking Assignments -- 18.10  
Design Scenario II: Example Using Non-blocking Assignments -- 18.11  
The 4-bit Register -- 18.12 Asynchronous Reset -- 18.13 Synchronous  
Reset -- 18.14 Design Guidelines and Summary -- 19 FPGA  
Architecture and Design Flow -- 19.1 Basics of Programmable Logic --  
19.2 CPLD Versus FPGA -- 19.3 ASIC Versus FPGA -- 19.4 FPGA  
Architecture -- 19.5 FPGA Design Flow -- 19.5.1 Design Planning --  
19.5.2 RTL Design -- 19.5.3 Design Verification and Synthesis --  
19.5.4 Design Implementation.  
19.5.5 Device Programming and Testing.

---