1. Record Nr. UNINA9910830781803321

Autore	Minns Peter D.
Titolo	Digital system design using FSMs : a practical learning approach / / Peter D. Minns
Pubbl/distr/stampa	Hoboken, NJ : , : John Wiley & Sons, Inc., , 2021
ISBN	1-119-78272-4 1-119-78271-6 1-119-78273-2
Descrizione fisica	1 online resource (340 pages)
Disciplina	621.381
Soggetti	Digital electronics Sequential machine theory
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Cover -- Title Page -- Copyright Page -- Contents -- Preface -- Acknowledgements -- About the Companion Website -- Guide to Supplementary Resources -- Chapter 1 Introduction to Finite State Machines -- 1.1 Some Notes on Style -- Chapter 2 Using FSMs to Control External Devices -- 2.1 Introduction -- Chapter 3 Introduction to FSM Synthesis -- 3.1 Introduction -- 3.2 Tutorials Covering Chapters 1, 2, and 3 -- 3.2.1 Binary data serial transmitter FSM -- 3.2.2 The high low FSM system -- 3.2.3 The clocked watchdog timer FSM -- 3.2.4 The asynchronous receiver system clocked FSM -- Chapter 4 Asynchronous FSM Methods -- 4.1 Introduction to Asynchronous FSM -- 4.2 Summary -- 4.3 Tutorials -- 4.3.1 FSM motor with fault detection -- 4.3.2 The mower in four and two states -- Chapter 5 Clocked One Hot Method of FSM Design -- 5.1 Introduction -- 5.2 Tutorials on the Clocked one Hot FSM Method -- 5.2.1 Seven-state system clocked one hot method -- 5.2.2 Memory tester FSM -- 5.2.3 Eight-bit sequence detector FSM -- Chapter 6 Further Event-Driven FSM Design -- 6.1 Introduction -- 6.2 Conclusions -- Chapter 7 Petri Net FSM Design -- 7.1 Introduction -- 7.2 Tutorials Using Petri Net FSM -- 7.2.1 Controlled shared resource Petri nets -- 7.2.2 Serial clock-driven Petri net FSM -- 7.2.3 Using asynchronous (event-driven) design with Petri nets -- 7.3 Conclusions

-- Appendix A1: Boolean Algebra -- A1.1 Basic Gate Symbols -- A1.2 The Exclusive OR and Exclusive NOR -- A1.3 Laws of Boolean Algebra -- A1.3.1 Basic OR rules -- A1.3.2 Basic AND rules -- A1.3.3 Associative and commutative laws -- A1.3.4 Distributive laws -- A1.3.5 Auxiliary rule for static 1 hazard removal -- A1.3.6 Consensus theorem -- A1.3.7 The effect of signal delay in logic gates -- A1.3.8 De-Morgan's theorem -- A1.4 Examples of Applying the Laws of Boolean Algebra -- A1.4.1 Converting AND-OR to NAND.
A1.4.2 Converting AND-OR to NOR -- A1.4.3 Logical adjacency rule -- A1.5 Summary -- Appendix A2: Use of Verilog HDL and Logisim to FSM -- A2.1 The Single-Pulse Generator with Memory Clock-Driven FSM -- A2.2 Test Bench Module and its Purpose -- A2.3 Using Synapticad Software -- A2.4 More Direct Method -- A2.5 A Very Simple Guide to Using the Logisim Simulator -- A2.5.1 The Logisim top level menu items -- A2.6 Using Flip-Flops in a Circuit -- A2.7 Example Single-Pulse FSM -- A2.8 How to Use the Simulator to Simulate the Single-Pulse FSM -- A2.8.1 Using Logisim with the truth table approach -- A2.9 Using Logisim with the Truth Table Approach -- A2.9.1 Useful note -- A2.10 Summary -- Appendix A3: Counters, Shift Registers, Input, and Output with an FSM -- A3.1 Basic Down Synchronous Binary Counter Development -- A3.2 Example of a Four-Bit Synchronous Up Counter with T Type Flip-Flops -- A3.3 Parallel Loading Counters - Using T Flip-Flops -- A3.4 Using D Flip-Flops To Build Parallel Loading Counters -- A3.5 Simple Binary Up Counter with Parallel Inputs -- A3.6 Clock Circuit to Drive the Counter (and FSM) -- A3.7 Counter Design Using Don't Care States -- A3.8 Shift Registers -- A3.9 Dealing with Input and Output Signals Using FSM -- A3.10 Using Logisim to Work with Larger FSM Systems -- A3.10.1 The equations -- A3.11 Summary -- Appendix A3: Counters, Shift Registers, Input, and Output with an FSM -- A4.1 Introduction -- A4.2 The Single-Pulse/Multiple-Pulse Generator with Memory FSM -- A4.3 The Memory Tester FSM Revisited -- A4.4 Summary -- Appendix A5: Programming a Finite State Machine -- A5.1 Introduction -- A5.2 The Parallel Loading Counter -- A5.3 The Multiplexer -- A5.4 The Micro Instruction -- A5.5 The Memory -- A5. 6 The Instruction Set -- A5.7 Simple Example: Single-Pulse FSM -- A5.8 The Final Example.
A5.9 The Program Code -- A5.10 Returning Unused States Via Other Transition Paths -- A5.11 Summary -- Appendix A6: The Rotational Detector Using Logisim Simulator with Sub-Circuits -- A6.1 Using the Two-State Diagram Arrangement -- Bibliography -- Index -- EULA.

Sommario/riassunto

"A finite state machine (FSM) is a computation model that can be implemented with hardware or software and can be used to simulate sequential logic and some computer programs. Finite state machines can be used to model problems in many fields including mathematics, artificial intelligence, games, and linguistics. This is a complete update of the author's earlier book, FSM-Based Digital Design using Verilog HDL (Wiley 2008). Whilst the essential foundation content remains, the book has been considerably refreshed to cover the design of Finite State Machines (FSM) in place of Microprocessors, using a novel form of State Machines based on Toggle Flip Flops (TFF) and Data Flip Flops (DFF). It follows a Linear Programmed Learning approach, enabling the reader to learn at their own pace, and to design their own FSM based systems."--