

Table of contents

- **1 Why Verilog Computer Design?**
- What is computer design?
- A brief history of computer/digital technology
- Translating algorithms into hardware
- Hardware description languages
- Typography
- Assumed background
- **2 Designing ASMs**
- What is an ASM chart?
- Pure behavioral example
- Mixed examples
- Pure structural example
- Hierarchical design
- **3 Verilog Hardware Description Language**
- Simulation versus synthesis
- Verilog versus VHDL
- Role of test code
- Behavioral features of Verilog
- Structural features of Verilog
- \$time
- Time control
- Assignment with time control
- Tasks and functions
- Structural Verilog, modules and ports
- **4 Three Stages for Verilog Design**
- Pure behavioral examples
- Mixed stage of the two-state division machine
- Pure structural stage of the two state division machine
- Hierarchical refinement of the controller
- **5 Advanced ASM Techniques**
- Moore versus Mealy
- Mealy version of the division machine
- Translating Mealy ASMs into behavioral Verilog
- Translating complex (goto) ASMs into behavioral Verilog
- Translating conditional command signals into Verilog
- Single-state Mealy ASMs
- **6 Designing for Speed and Cost**
- Propagation delay
- Factors that determine clock frequency
- Example of netlist propagation delay
- Abstracting propagation delay
- Single cycle, multi-cycle and pipeline
- **7 One Hot Designs**
- Moore ASM to one hot

- Verilog to one hot
- Mealy commands in a one hot machine
- Moore command signals with Mealy
- 8 General-Purpose Computers
- Introduction and history
- Structure of the machine
- Behavioral fetch/execute
- Mixed fetch/execute
- Memory hierarchy
- 9 Pipelined General-Purpose Processor
- First attempt to pipeline
- Example of independent instructions
- Data dependencies
- Data forwarding
- Control dependencies: implementing JMP
- Skip instructions in a pipeline
- Our old friend: division
- Multi-port memory
- Pipelined PDP-8 architecture
- 10 RISC Processors
- History of CISC versus RISC
- The ARM
- Princeton versus Harvard architecture
- The register file
- Three operands are faster than one
- ARM subset
- Multi-cycle implementation of the ARM subset
- Pipelined implementation
- Superscalar implementation
- Comparison of childish division implementations
- 11 Synthesis
- Overview of synthesis
- Verilog synthesis styles
- Synthesizing enabled_register
- Synthesizing a combinational adder
- Synthesizing an implicit style bit serial adder
- Switch debouncing and single pulsing
- Explicit style switch debouncer
- Putting it all together: structural synthesis
- A bit serial PDP-8
- A Machine and Assembly Language
- B PDP-8 Commands
- Memory reference instructions
- Non-memory reference instructions
- Group 1 microinstructions
- Group 2 microinstructions

- **C Combinational Logic Building Blocks**
- **Models of reality**
- **Bus**
- **Adder**
- **Multiplexer**
- **Other arithmetic units**
- **Arithmetic logic unit**
- **Comparator**
- **Demux**
- **Decoders**
- **Encoders**
- **Programmable devices**
- **Conclusions**
- **Further reading**
- **Exercises**
- **D Sequential Logic Building Blocks**
- **D.1 System clock**
- **Timing Diagrams**
- **Synchronous Logic**
- **Bus timing diagrams**
- **The D-type register**
- **Enabled D-type register**
- **Up counter register**
- **Up/down counter**
- **Shift register**
- **Unused inputs**
- **Highly specialized registers**
- **Further Reading**
- **Exercises**
- **E Tri-State Devices**
- **Switches**
- **Single bit tri-state gate in structural Verilog**
- **Bus drivers**
- **Uses of tri-state**
- **Further Reading**
- **Exercises**
- **F Tools**