- Apply these concepts to perform computer arithmetic

- Express, simplify, and minimize Boolean functions, using truth tables, Boolean algebra,
- Utilize technology mapping techniques, such as NAND only
- Use decoders and multiplexers to implement a

- Represent sequential systems using finite state
- Produce state diagrams, state tables, excitation tables and state equations, including one
- Use timing diagrams to explain basic sequential circuit timing issues

- Analyze and design synchronous sequential logic

- Analyze and design circuits using Datapath components: registers, adders, shifters, multipliers
- Analyze and design multi-cycle datapaths
- Explain arithmetic logic units (ALUs), register files, memory
- Explain elementary register transfer level designs
- Describe the organization and operation of a basic digital computer

- Describe the difference between HDL and software
- Develop synthesizable HDL
- Verify designs using a testbench
- Use behavioral models of logic circuits
- Map HDL models to logic circuits on field programmable gate arrays (FPGAs)

- Explain the hierarchical relationship between transistors, digital logic, and microprocessors
- Explain the operation of machine language vis a vis computer architecture
- Analyze and revise basic assembly language computer programs using a 5-stage pipelined processor

- Implement and debug combinational and sequential logic circuits using standard logic chips
- Implement and debug combinational and sequential logic circuits using HDL and FPGAs
- Implement and debug basic systems using microcontrollers

- Design, simulate, build, validate, debug, and document digital circuits

- x Number systems
- x Logic gates
- x Boolean algebra
- x Datapath Components
- x Hardware Description Language (HDL)
- x Introduction to Assembly Language
- x Pipelining
- x Design process