

Intel® FPGA Technical Training

Quartus Prime Tcl Scripting

Course Description

This course presents the Tcl scripting capabilities in the Quartus Prime software.

The course begins by introducing the industry-standard Tcl scripting language supported by Quartus Prime to customize functionality or automate tasks. It covers syntax and basic data types such as strings, lists, and arrays.

The course continues by covering commonly used Quartus Prime software Tcl packages and four common uses of Tcl scripting in the Quartus Pro compilation flow, with examples.

The course also covers useful Tcl commands in the Quartus Pro software, accessing command-line options, and running Tcl scripts.

The course combines 50% theory with 50% practical work.

Course Duration

2 days

Goals

1. Be familiar with common Tcl commands
2. Be familiar with basic rules of Tcl syntax
3. Read, write and run Tcl scripts
4. Understand how to use Tcl control structures
5. Write Tcl procedures
6. Use Tcl commands for file I/O
7. Understand Tcl scripting supported in Quartus Prime software
8. Write scripts to create projects, compile designs, extract report data, and run timing analysis reports
9. Understand useful Quartus Prime software-specific Tcl commands
10. Write TCL script files to automate constraining and analysis of FPGA designs



When innovation meets expertise...

Intended Users

Hardware engineers who develop FPGAs and would like to automate processes within Quartus Prime software

Previous Knowledge

FPGA design

TimeQuest

Course Material

1. Quartus Prime software
2. Course book (including labs)



When innovation meets expertise...

Table of Contents

Day #1

❖ Introduction to Tcl

- Introduction to Tcl Scripting
 - About Tcl
 - Tcl language basics
 - Tcl interpreter basics
 - Hello World example
 - Running Tcl commands
 - Creating and running Tcl scripts
- Syntax & Operators
 - Variables
 - Substitution
 - Variable value substitution
 - Command substitution
 - Backslash substitution
 - Grouping
 - Comments
 - expr command
 - Arithmetic grouping and precision
 - incr command
 - Arithmetic operators
 - Logical operators
- Tcl Data Types
 - Strings
 - String commands
 - Glob-style matching
 - Strings and conditional tests
 - Lists
 - List commands
 - Arrays
 - Array commands
- Control Structures
 - if statements
 - elsif keyword
 - switch statements
 - for statements
 - foreach command
 - for and foreach comparison
 - while command
 - Loop control
 - Error handling



When innovation meets expertise...

- Procedures & Scope
 - Procedures
 - proc command
 - Scope
 - upvar command

- File I/O Commands
 - File I/O
 - Opening files
 - Writing files
 - Reading files
 - Using command-line arguments

- ❖ **Lab #1:** Write your first Tcl script
- ❖ **Lab #2:** Use advanced language features to create more complex scripts

Day #2

❖ Quartus Prime Tcl Scripting

- Quartus Prime Tcl Support
 - Tcl packages
 - Commonly used packages

- Creating Projects
 - ::quartus::project package
 - Saving assignments to QSF
 - export_assignments scenario
 - Examples

- Compiling Designs
 - ::quartus::flow package
 - Project clean
 - Examples

- Accessing Report Data
 - ::quartus::report package
 - Report structure
 - Panel structure
 - Examples

- Timing Analysis and Tcl
 - Timing analysis Tcl packages
 - Running TimeQuest from command line
 - Running timing analysis during flow

- SDC and Tcl examples
 - Miscellaneous Quartus Tcl Scripting Commands
 - ::quartus::misc package
 - Escaping brackets
 - escape_brackets command
 - post_message command
 - Tcl collections
 - Accessing Command-Line Options
 - Using argv, argc, and argv0
 - Processing command-line options
 - Using Tcl in the Quartus Prime Software
 - Using Tcl in Quartus Prime
 - Using Tcl at the command line
 - Running Tcl Scripts
 - Why different executables?
 - Quartus array
 - Generate Tcl scripts for project
 - QHelp utility
 - Qsys related Tcl uses
- ❖ **Lab #3:** Write Tcl script to automate project in Quartus Prime software
- ❖ **Lab #4:** Write Tcl script to analyze timing in TimeQuest