

FPGAs for Embedded Systems

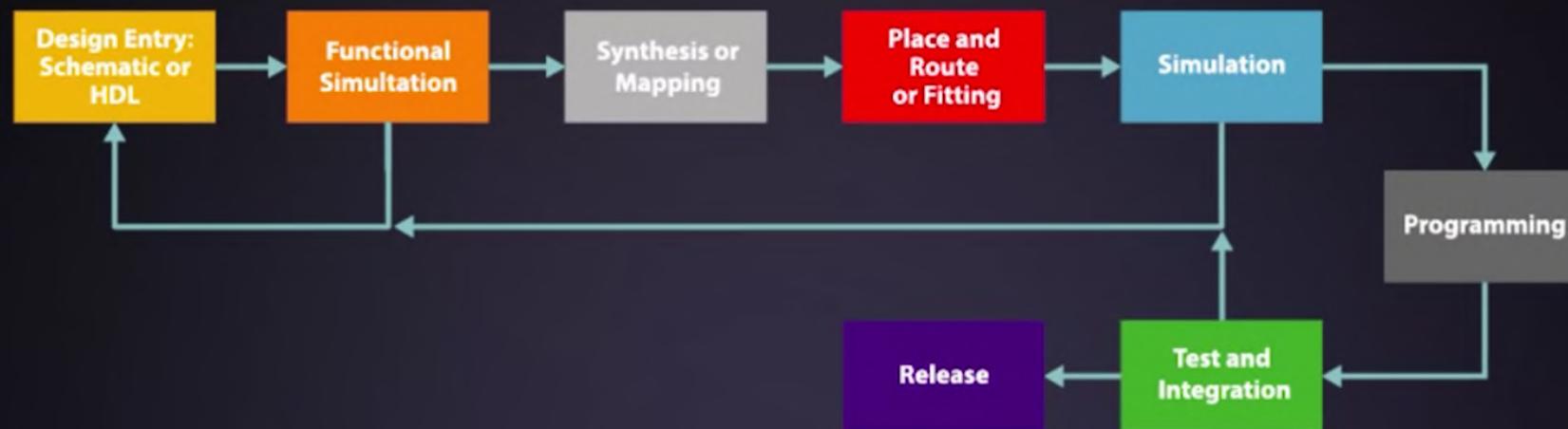
Module 1

Programmable Logic
Devices and the FPGA

Module 2

Use Quartus Prime for
sample FPGA Design

The FPGA Design Flow



Design Entry



Quartus Prime provides a number of methods for schematic entry, including:

- Schematic Capture
- Import of IP blocks
- HDL text entry, including VHDL, Verilog, and System - Verilog
- State machine entry
- Import of EDIF files

Design Analysis in Time



Quartus Prime's Analysis and Synthesis module:

- checks the design source files for errors,
- builds a database
- synthesizes and optimizes the logic design and
- maps the design logic to device resources.

Design Fitting (Compiling a Design)



Fitting the design in the smallest possible part is a principal design challenge in FPGA design.

Quartus Prime allows control of the fitting process by choice of fitting approaches, including:

Balanced

High Performance (speed)

Low Power

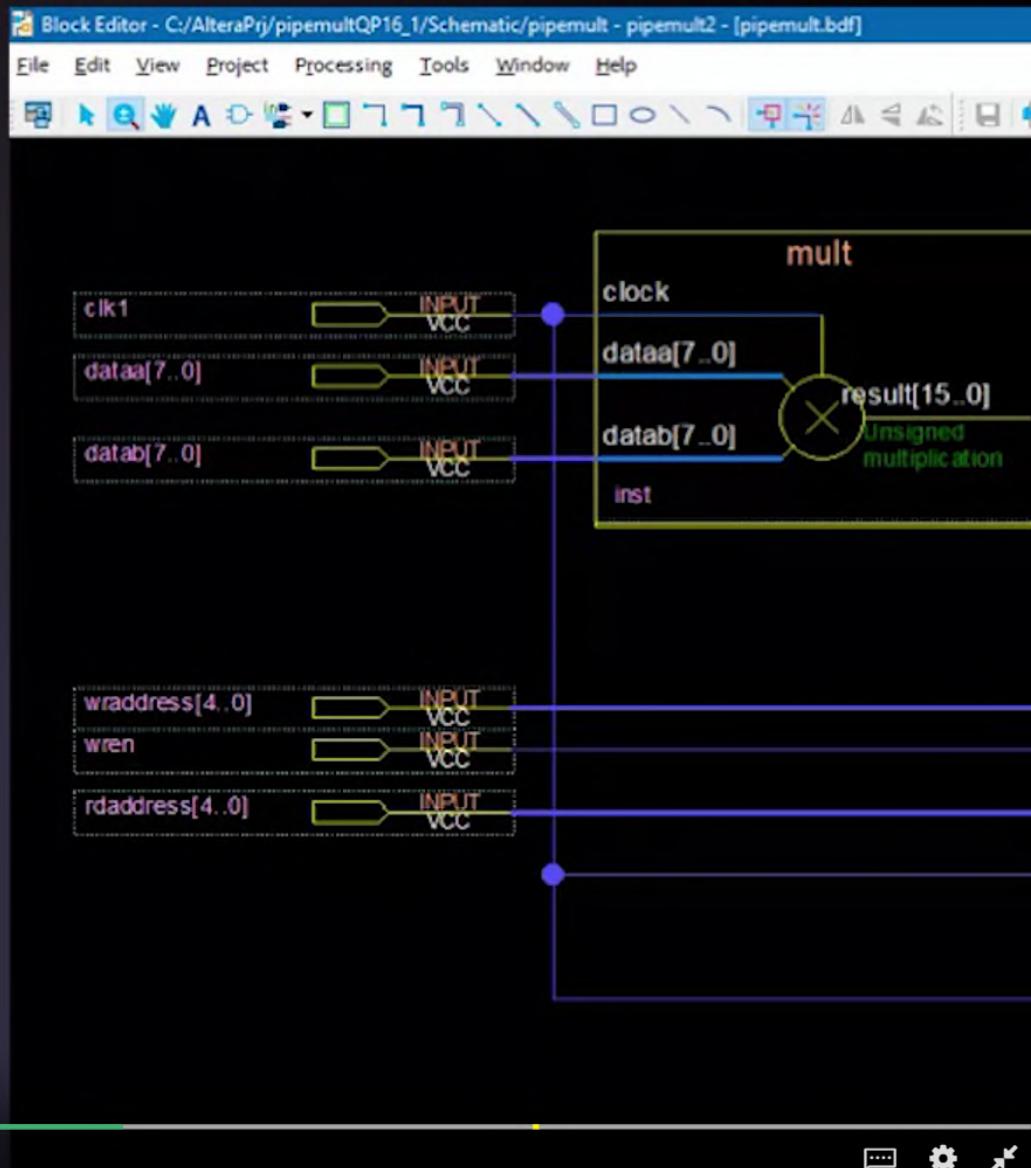
Small Area

Design Analysis in Pictures

Quartus Prime can provide several views of a design in pictures, representing different aspects of the design.

Picture representations of the design from the RTL Viewer, or the Technology Map Viewer, can give you confidence in your design implementation and design techniques.

The Chip Planner provides a picture of the chip showing the use of the design resources.



Design Analysis in Time



Synchronization is fundamental to reliable FPGA designs.
The synchronizing signal is the clock.

Static Timing Analysis can determine if there are violations of timing requirements relative to the clock.

The TimeQuest timing analyzer uses a set of equations to calculate slack, and also to determine Fmax.

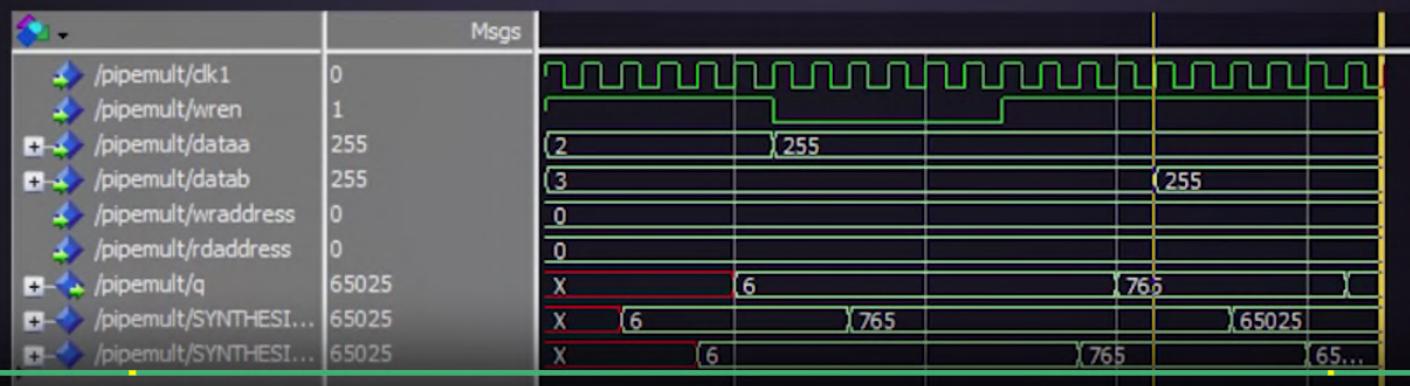
We will learn how to write effective constraints to improve Fmax.

Design Verification by Simulation



ModelSim is a powerful simulation tool integrated with Quartus Prime. It can create accurate timing diagrams of all the signals in a design, as well as stimulus signals for testing.

We will use this tool to test and better understand the operation of our example design.



Videos in Module 2

1. The FPGA Design Flow
2. Downloading Quartus Prime
3. Installing Quartus Prime
4. Introducing Quartus Prime
5. Create a design project in Quartus Prime
6. Create a top-level design in Quartus Prime
7. Compile a design
8. View the RTL
9. Timing Analysis with TimeQuest: Part I
10. Timing Analysis with TimeQuest: Part II
11. Simulate an FPGA Design with ModelSim

This presentation will help you install Altera's state of the art FPGA design tool, **Quartus Prime**.

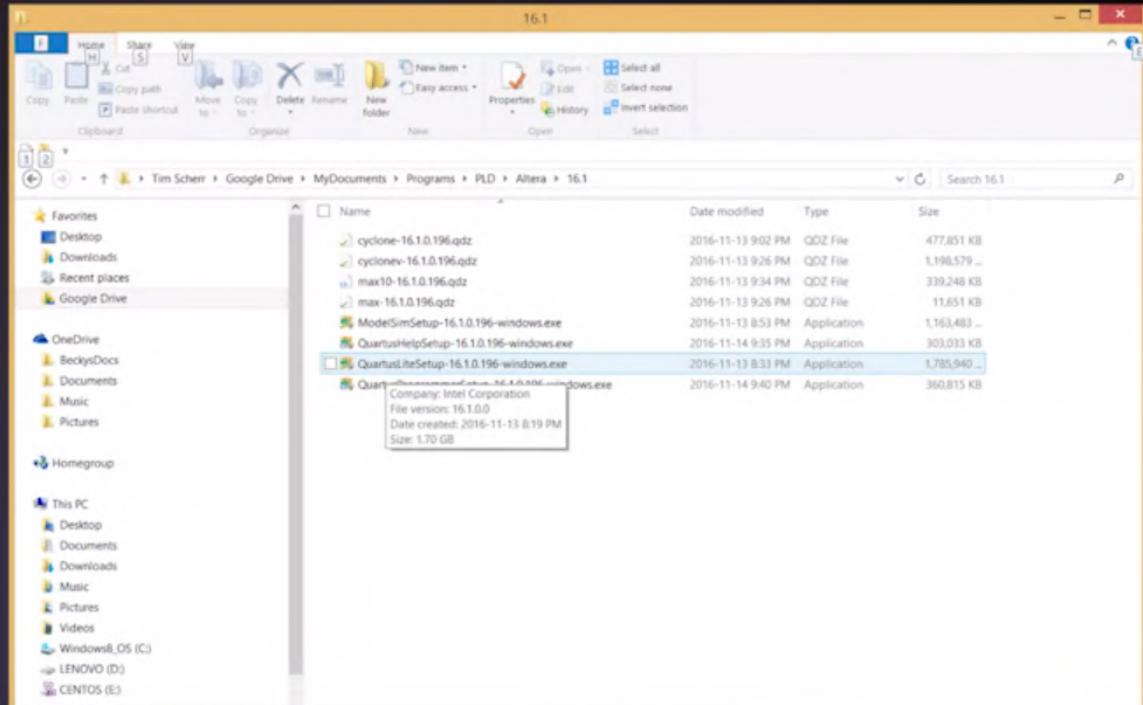
This is a very powerful tool that we will use throughout the course to explore the world of FPGA designs.

There are a number of steps in the installation; please follow them carefully to avoid any frustration.



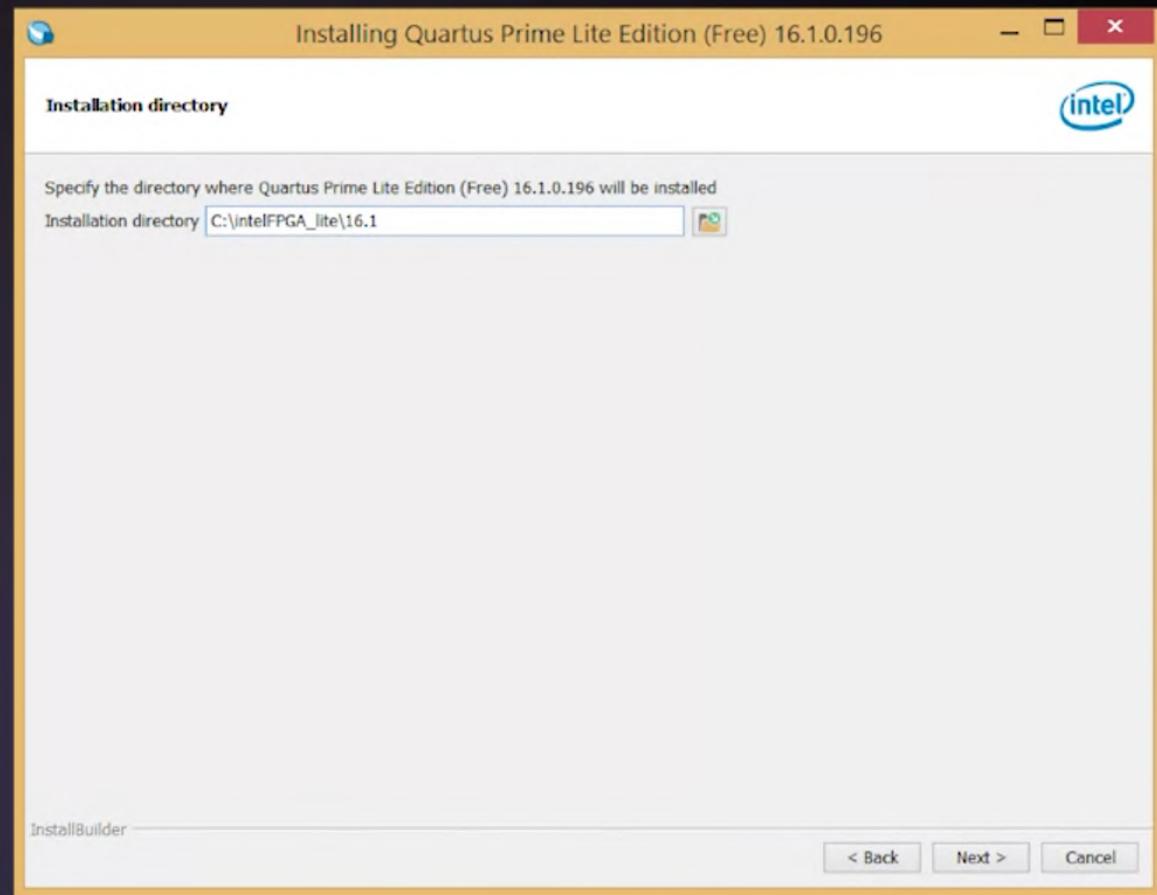
Feel free to pause this video while the programs proceed to install.

Begin the installation
by selecting the
QuartusLiteSetup.exe

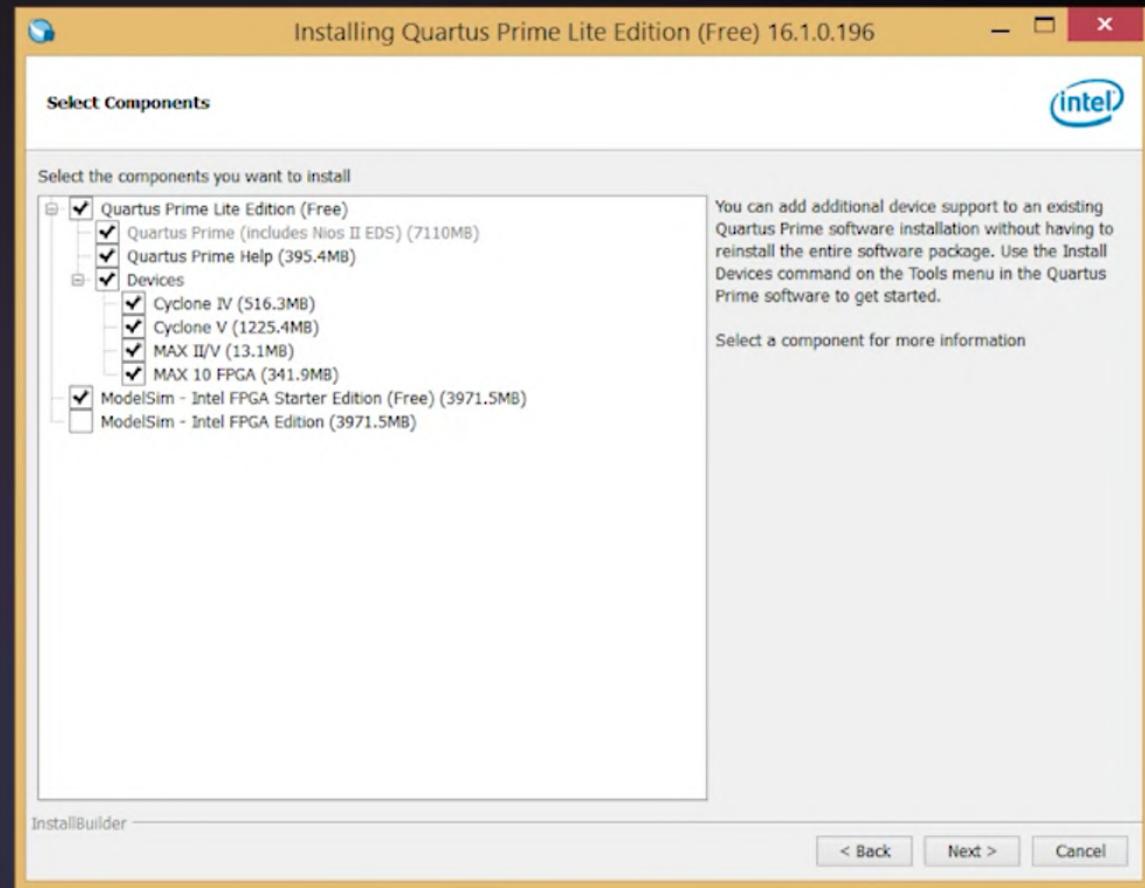


Click through the installation dialog boxes.

Use the default installation directory for best results.

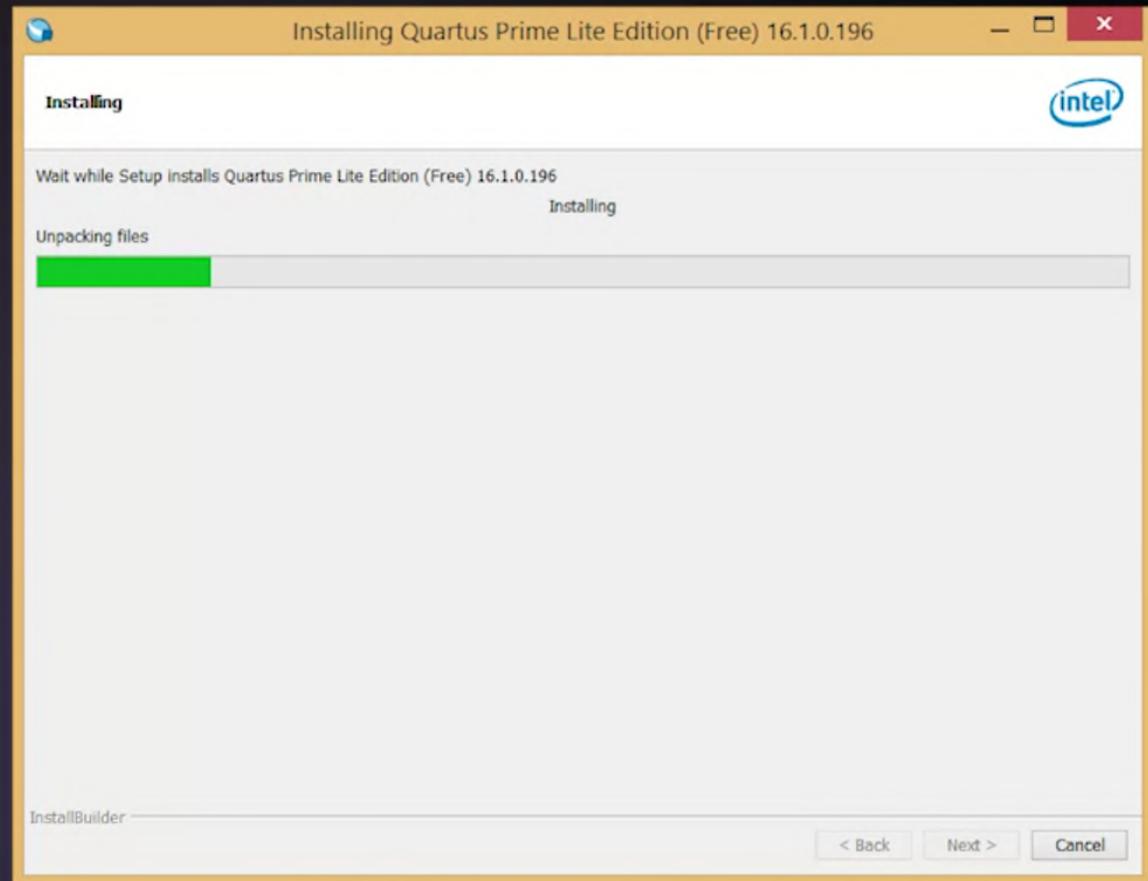


When you get to the select components screen, leave all components checked.

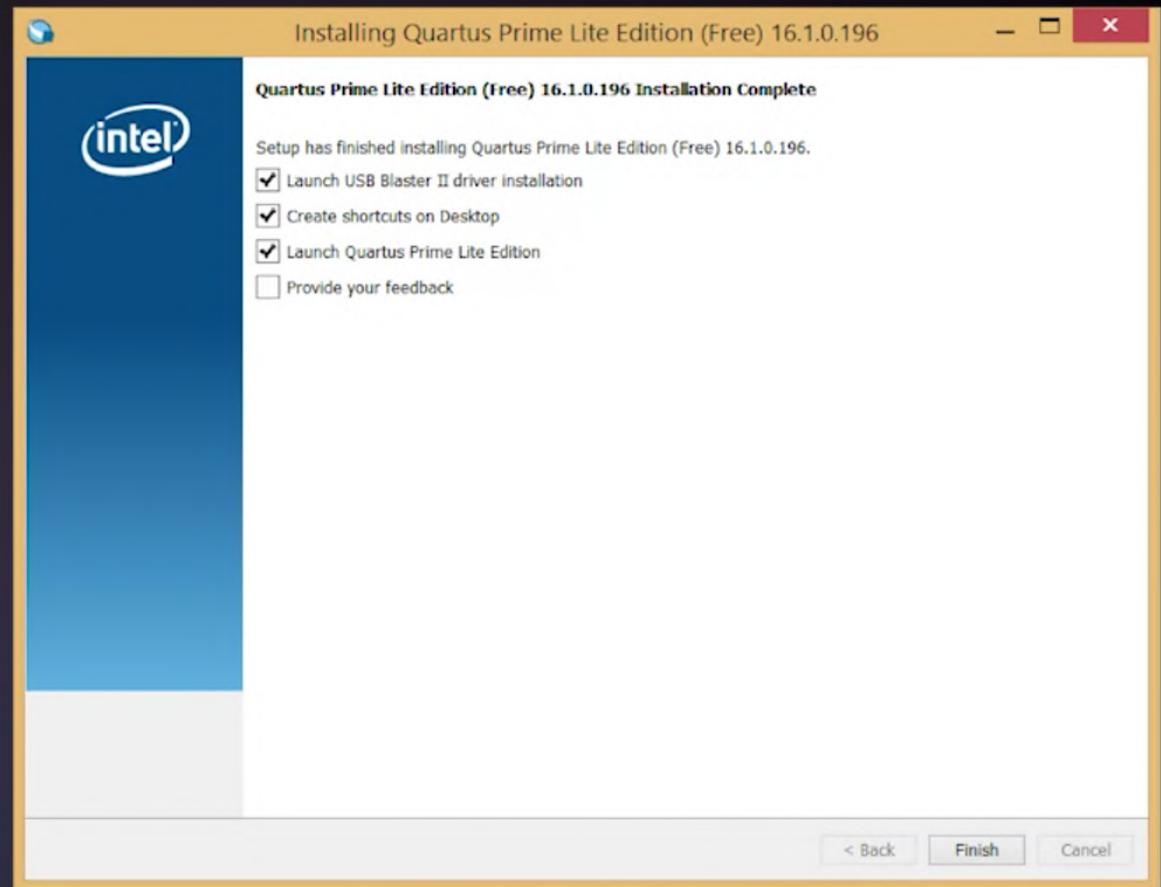


Hit **next** and the installation will begin.

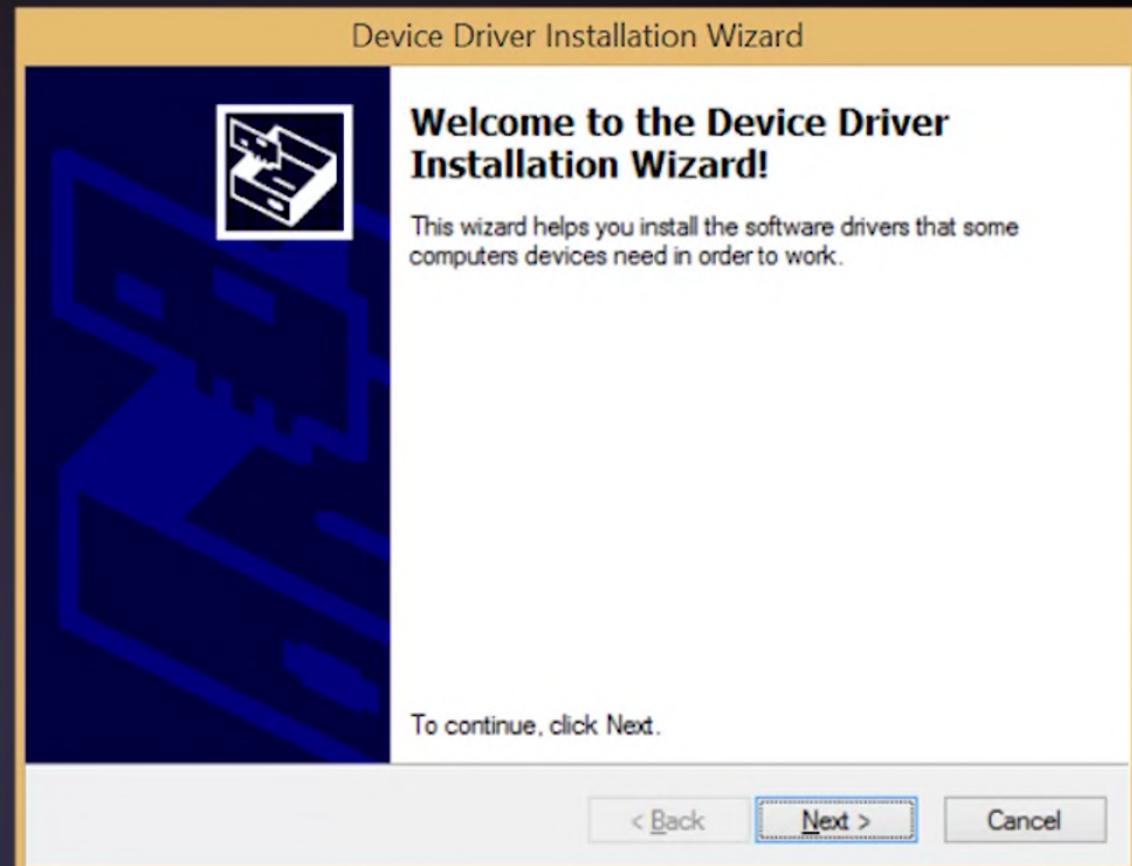
Required hard drive space: 14 gigabytes



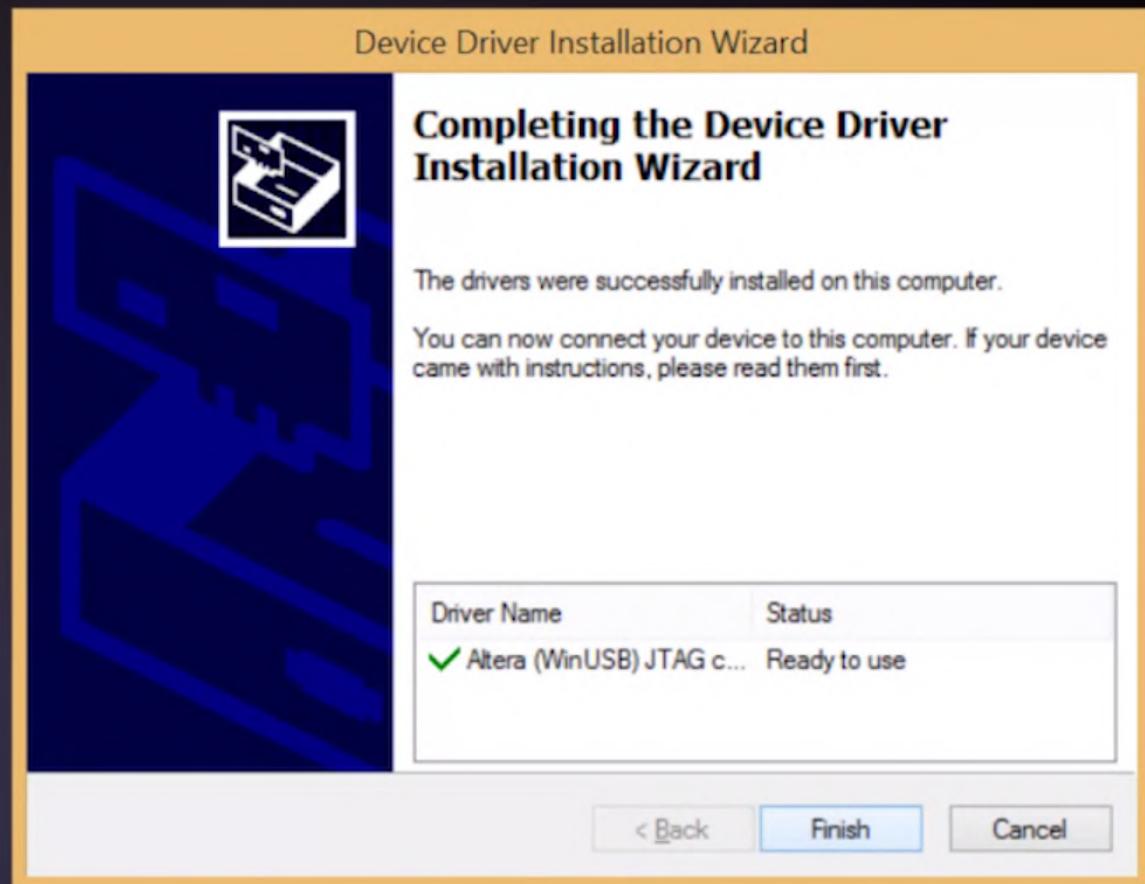
Once installation is complete, hit **finish** to start.



When complete, a device driver installation may run.



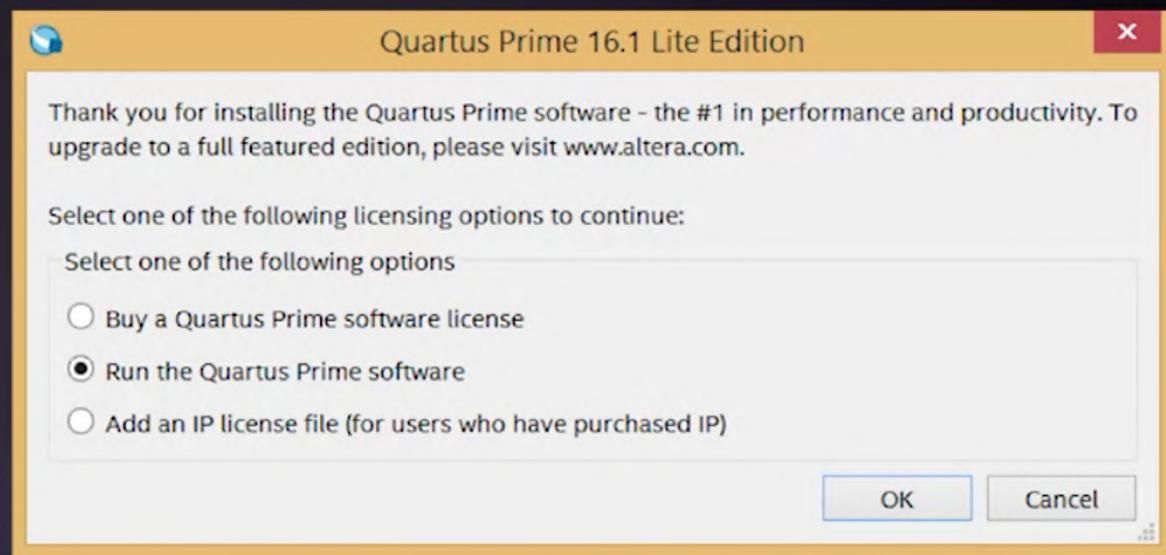
Allow this to finish and
you are done with the
Quartus Prime
installation!



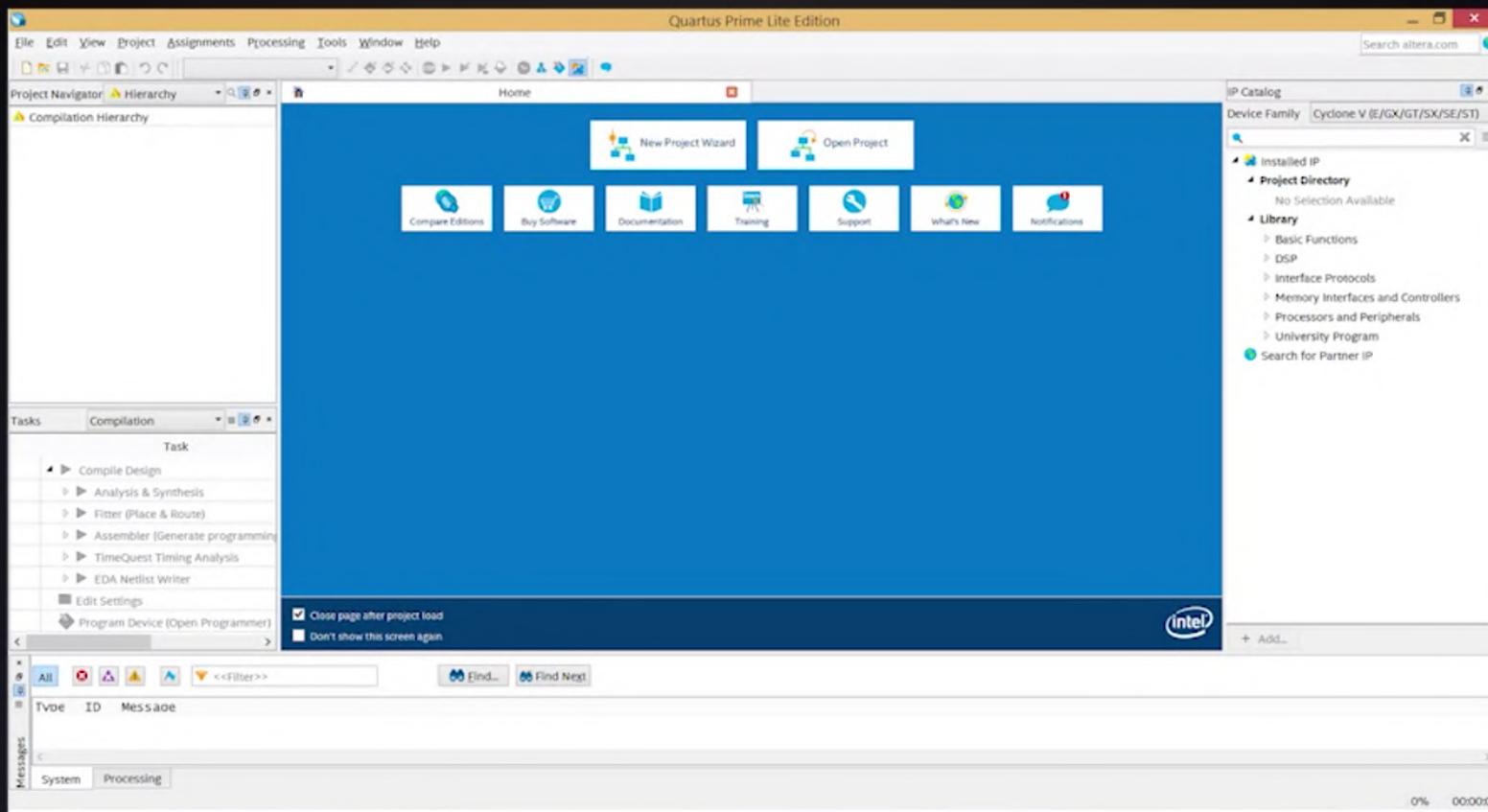
You may see a dialog that gives you the opportunity to purchase Quartus.

Do not select that (unless you want to).

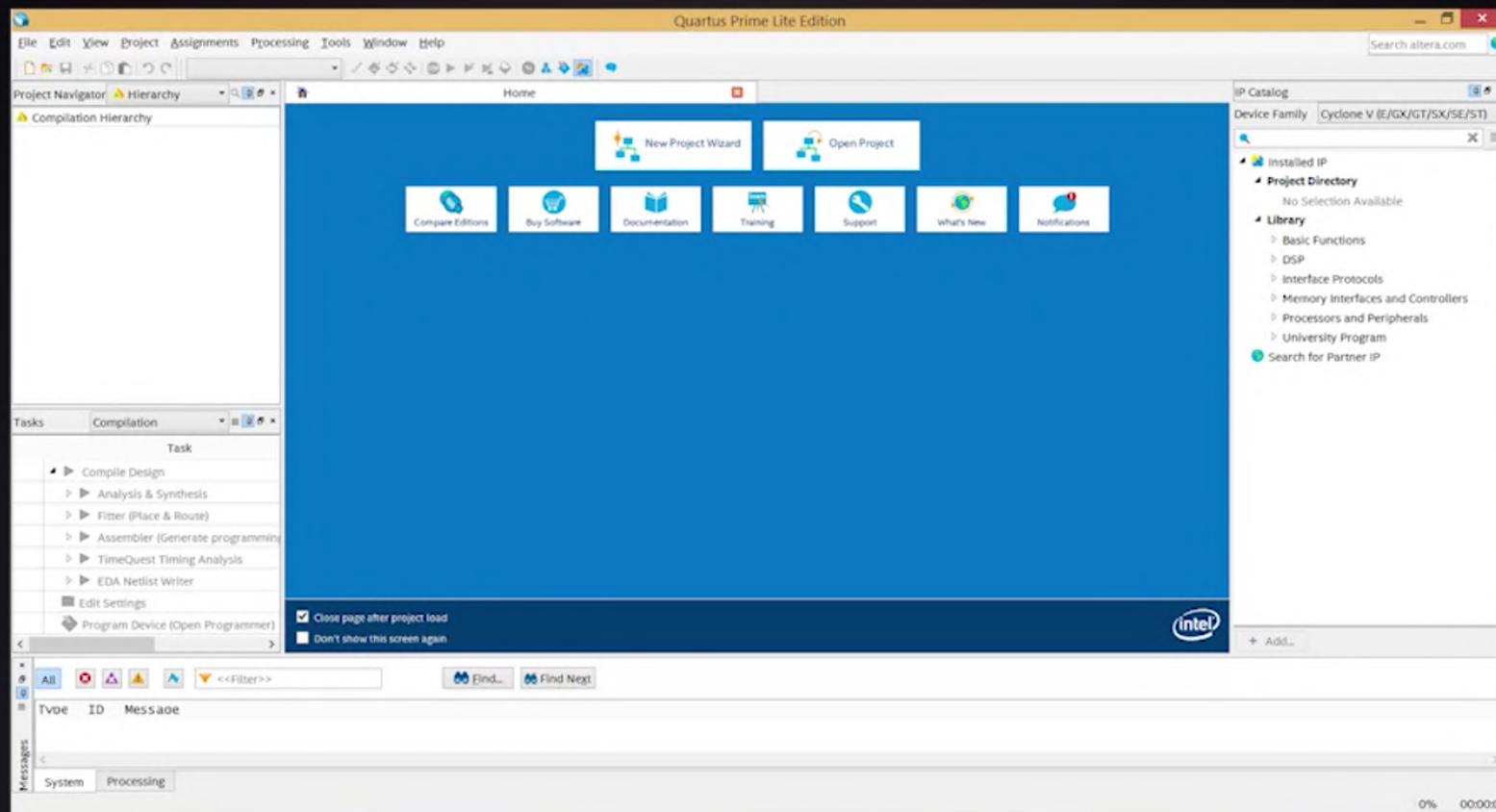
Instead, select **Run Quartus Prime**.



You should see this window as Quartus Prime runs.



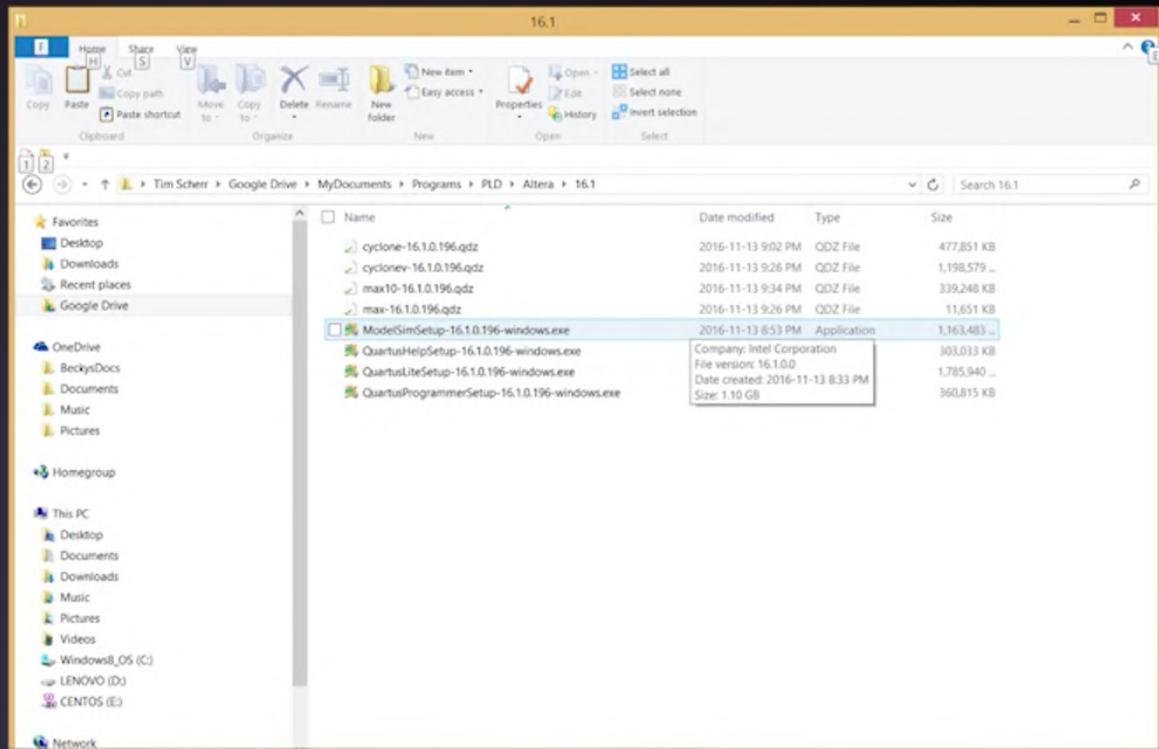
If not, click on the Quartus Prime desktop icon



or on Intel FPGA - Quartus Prime in your start menu.

Before you use
Quartus Prime, you
need to install Quartus
Help, and the Quartus
Programmer.

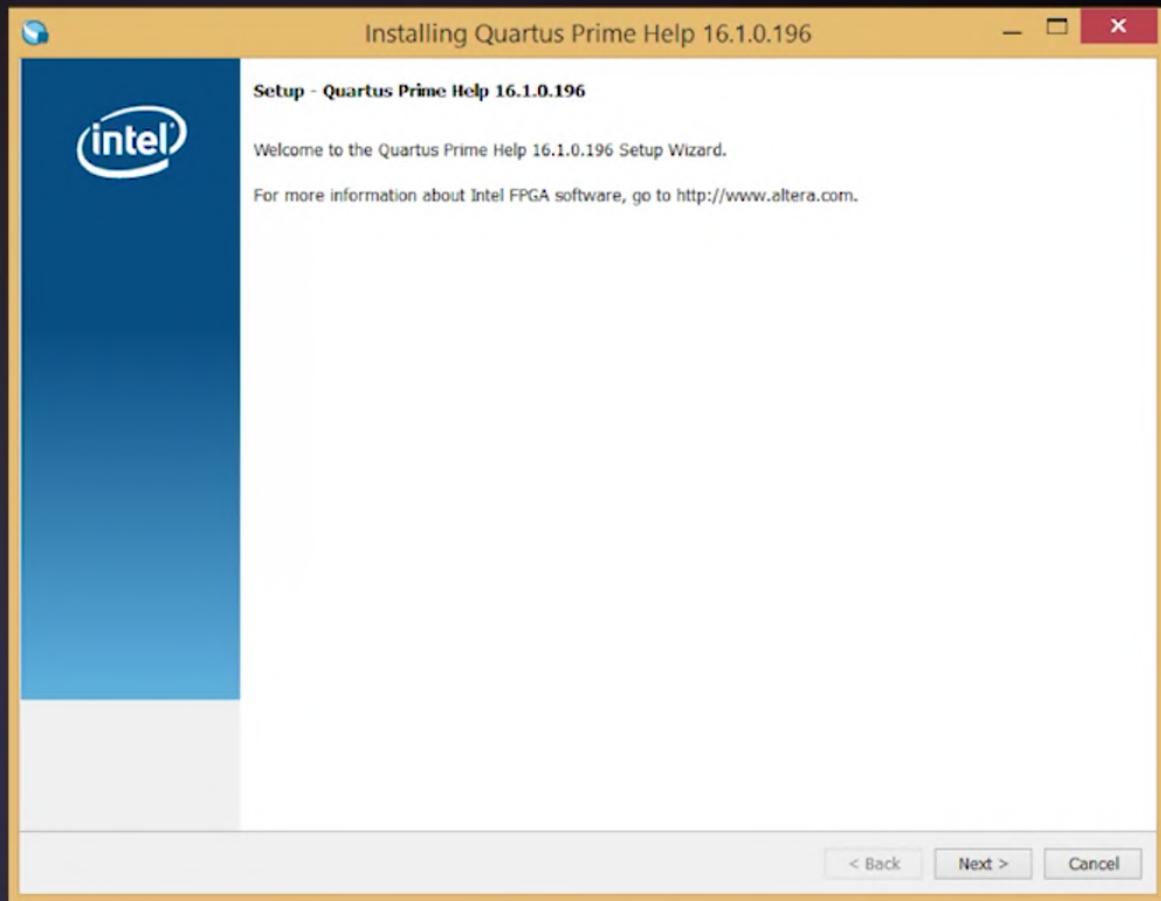
Next Up:
Quartus Help



After starting the Quartus Help Setup.exe, you should see this setup screen.

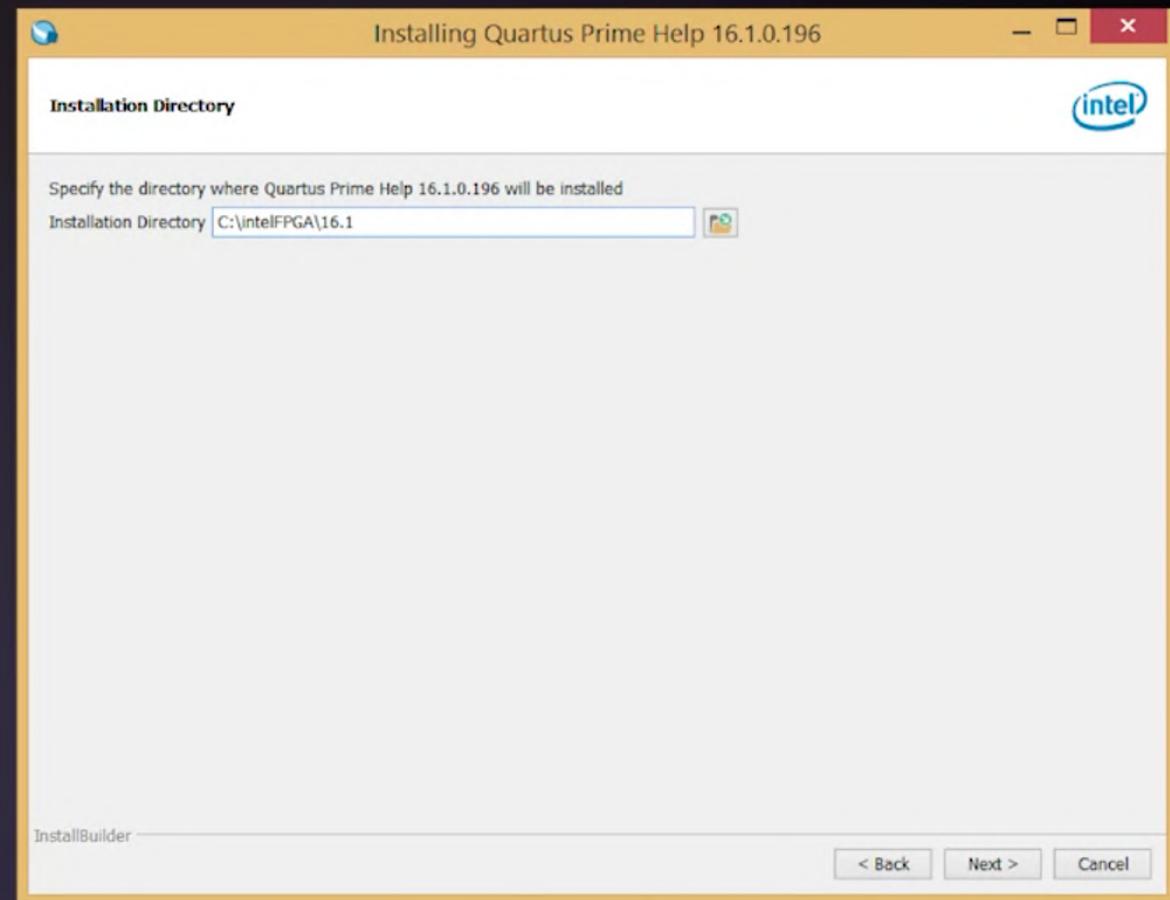
Click **Next**.

Accept the software license on the next screen, then click **Next**.

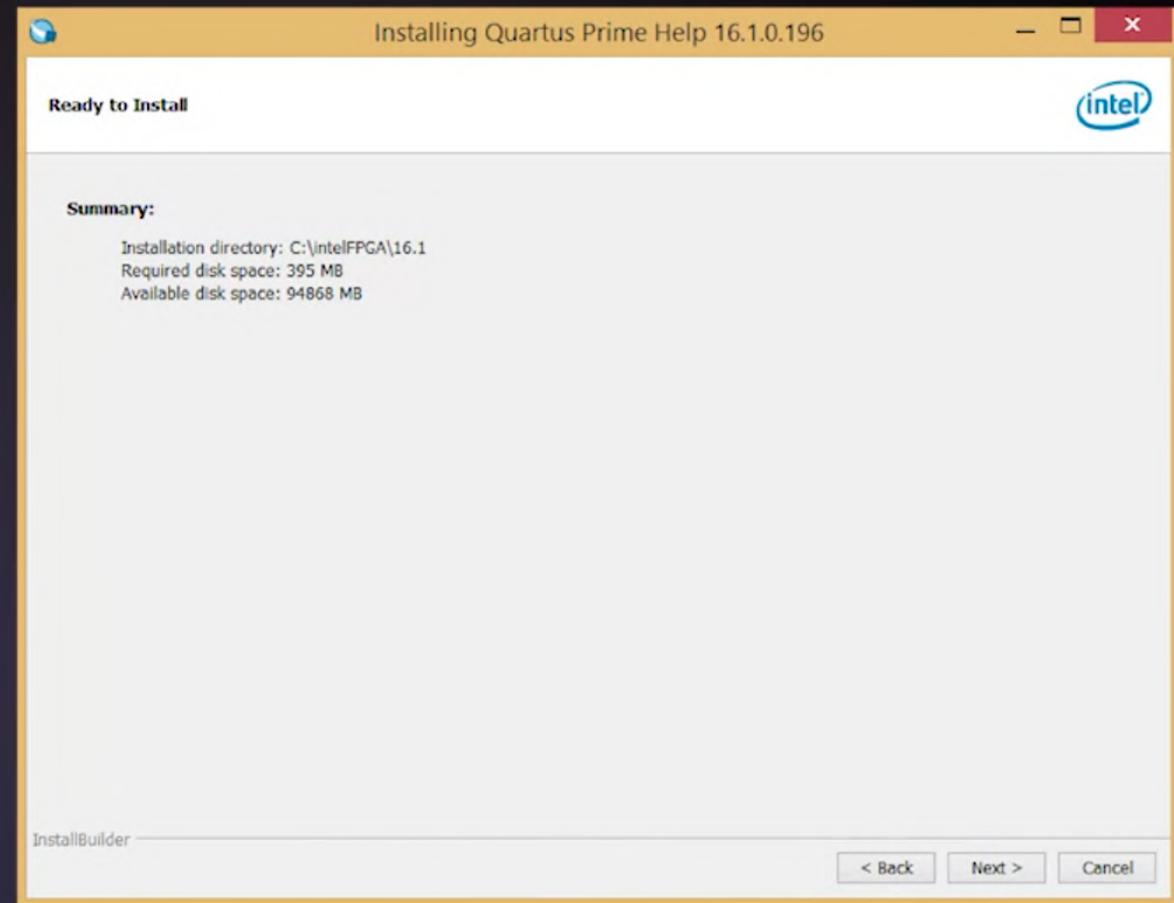


Choose a directory to install Quartus Prime Help.

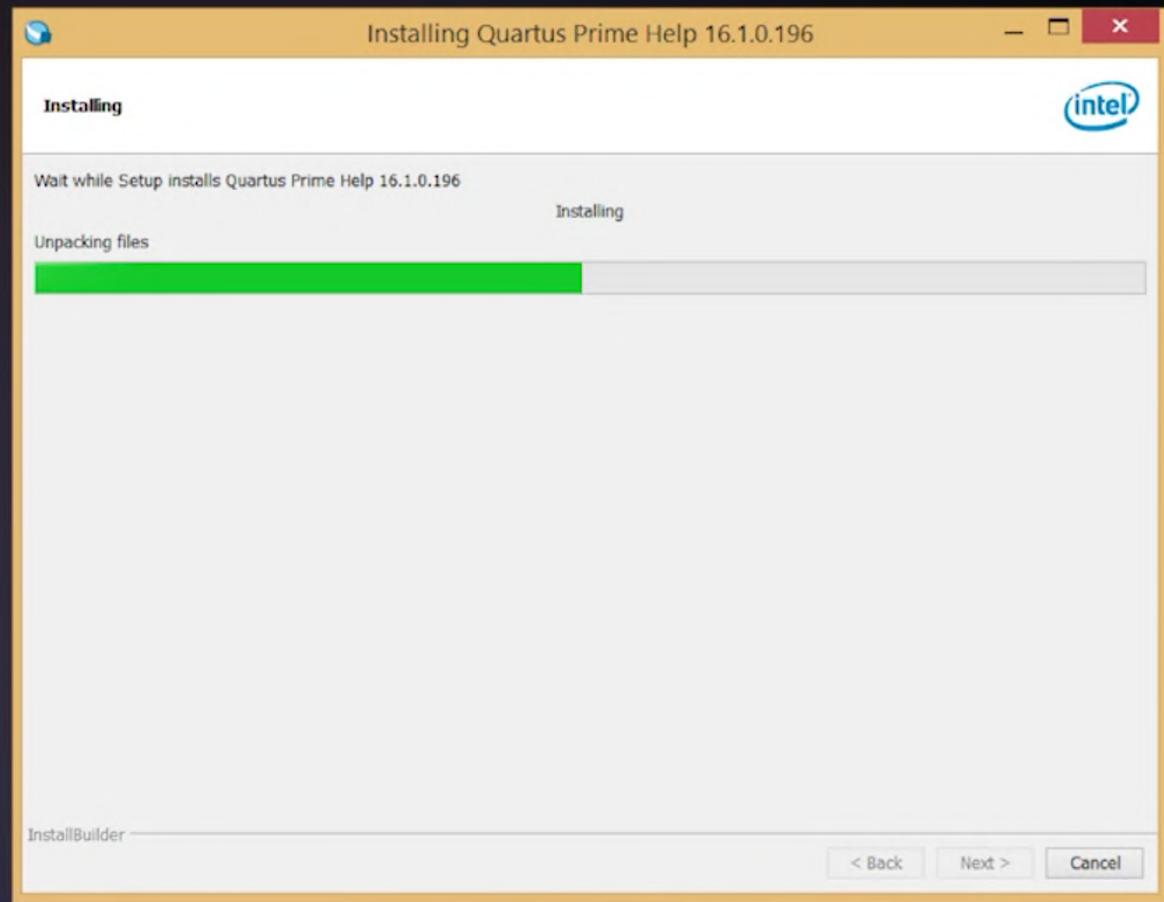
The default is usually the safest.



Now you are ready to install, provided you have enough memory.



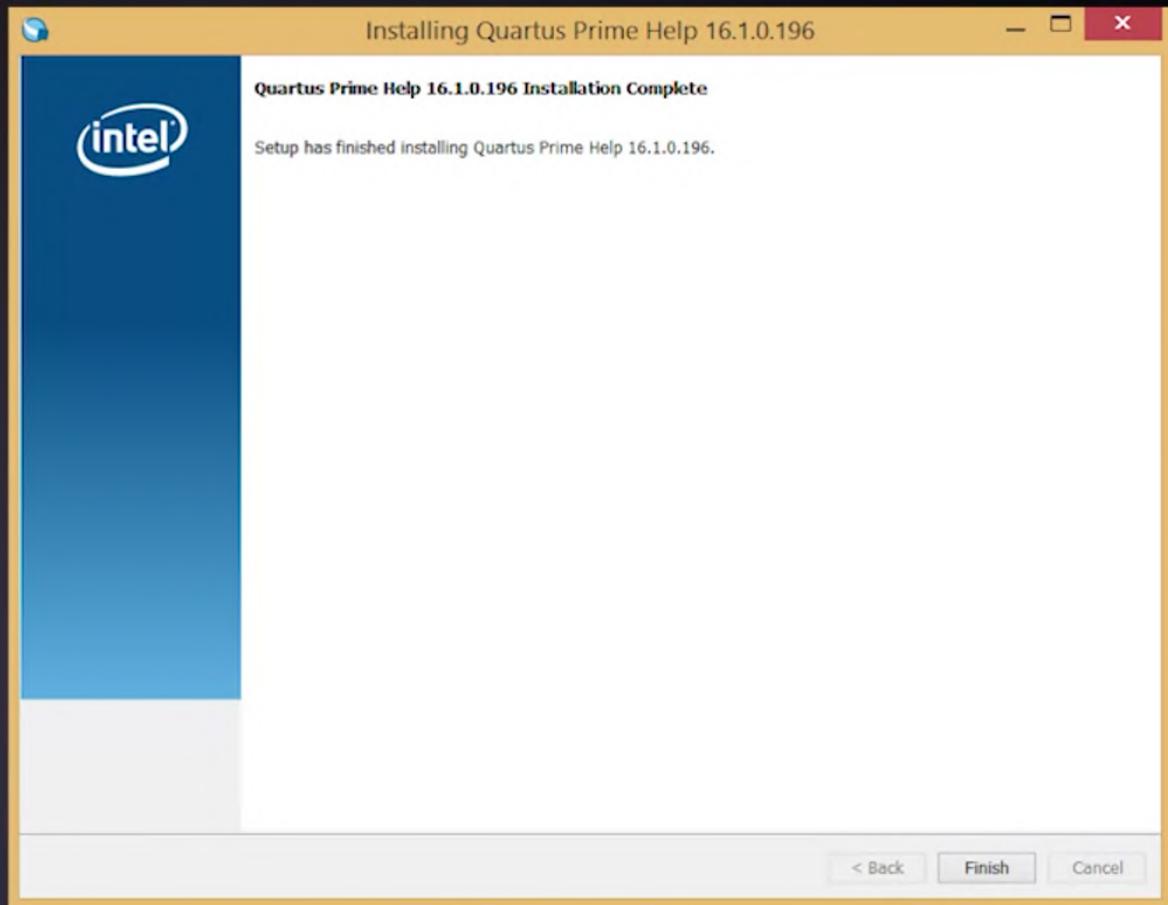
Installing...



Done!

You are finished with
the Quartus Help
installation.

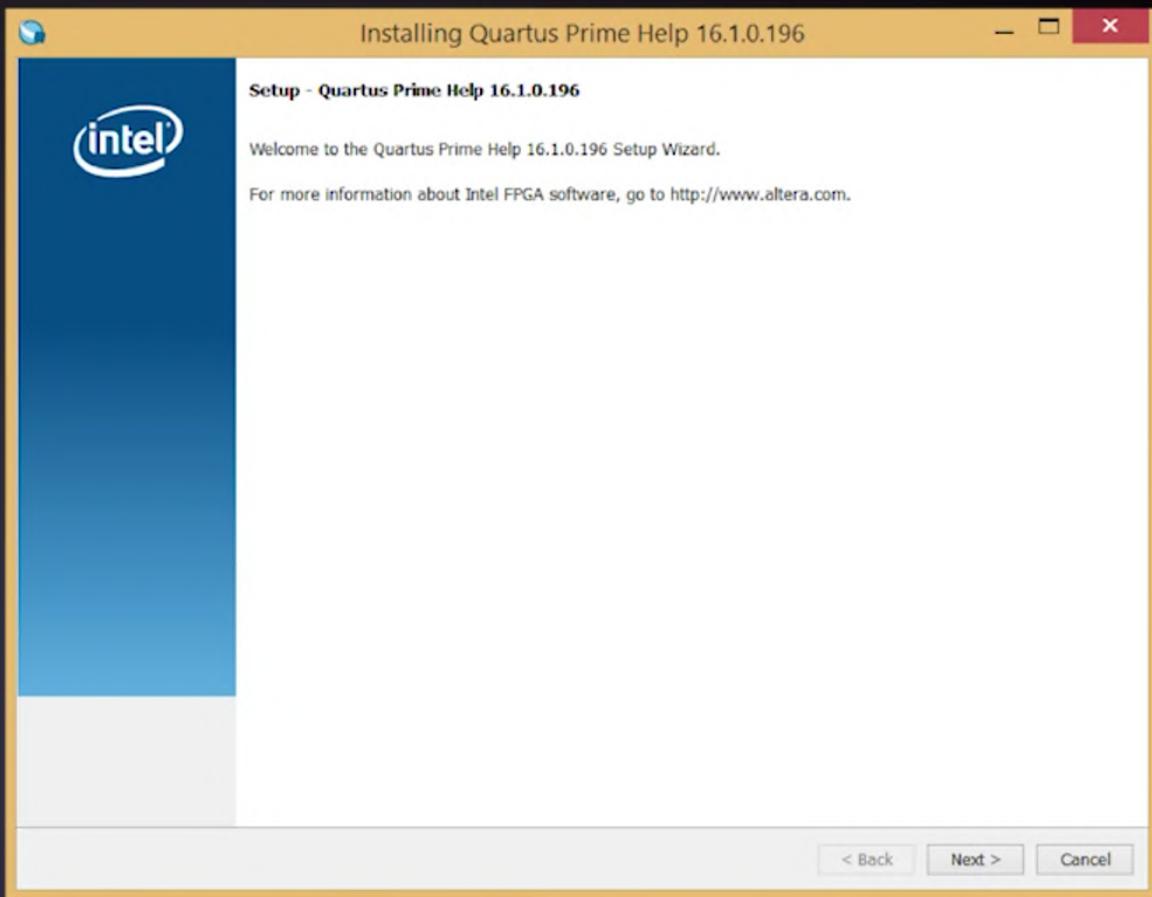
Click **Finish**.



After starting the Quartus Programmer setup.exe, you should see this setup screen.

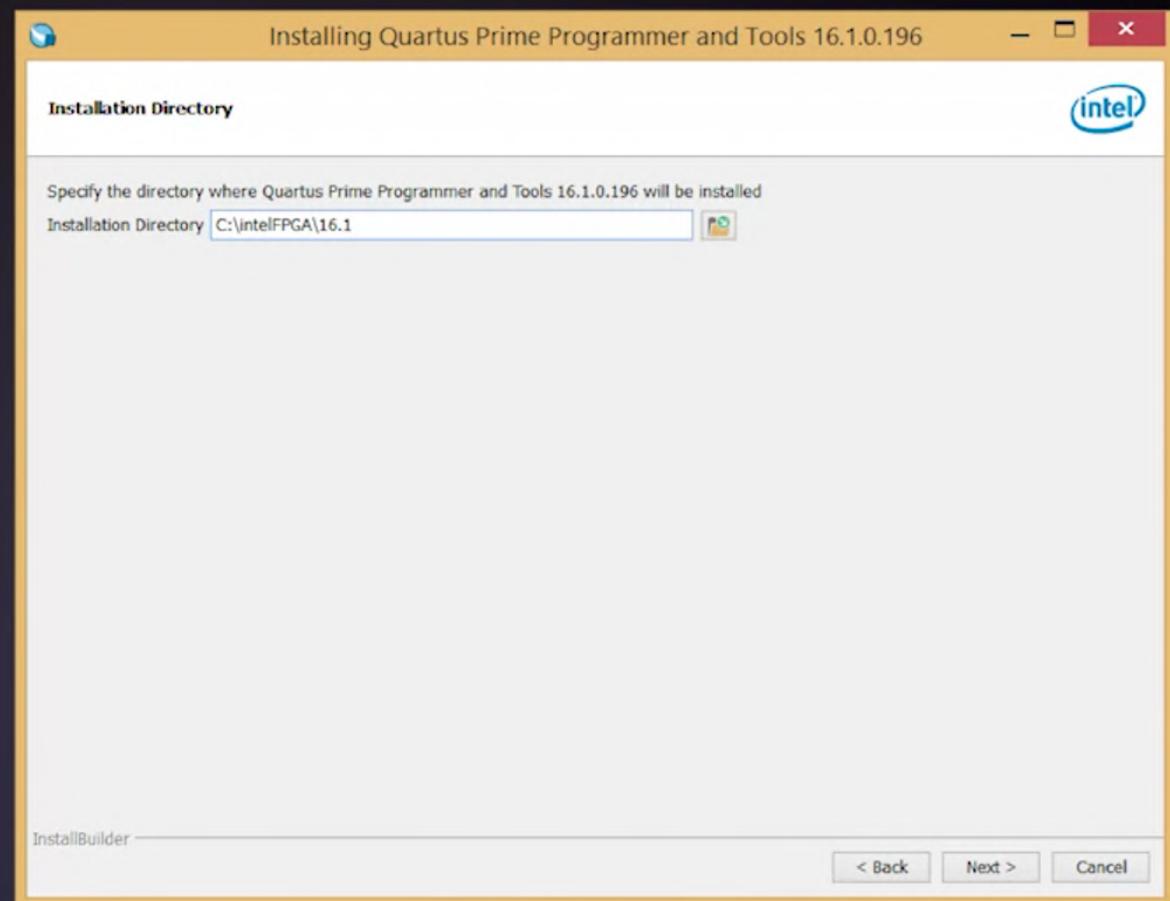
Click **Next**.

Accept the software license on the next screen, and click **Next**.

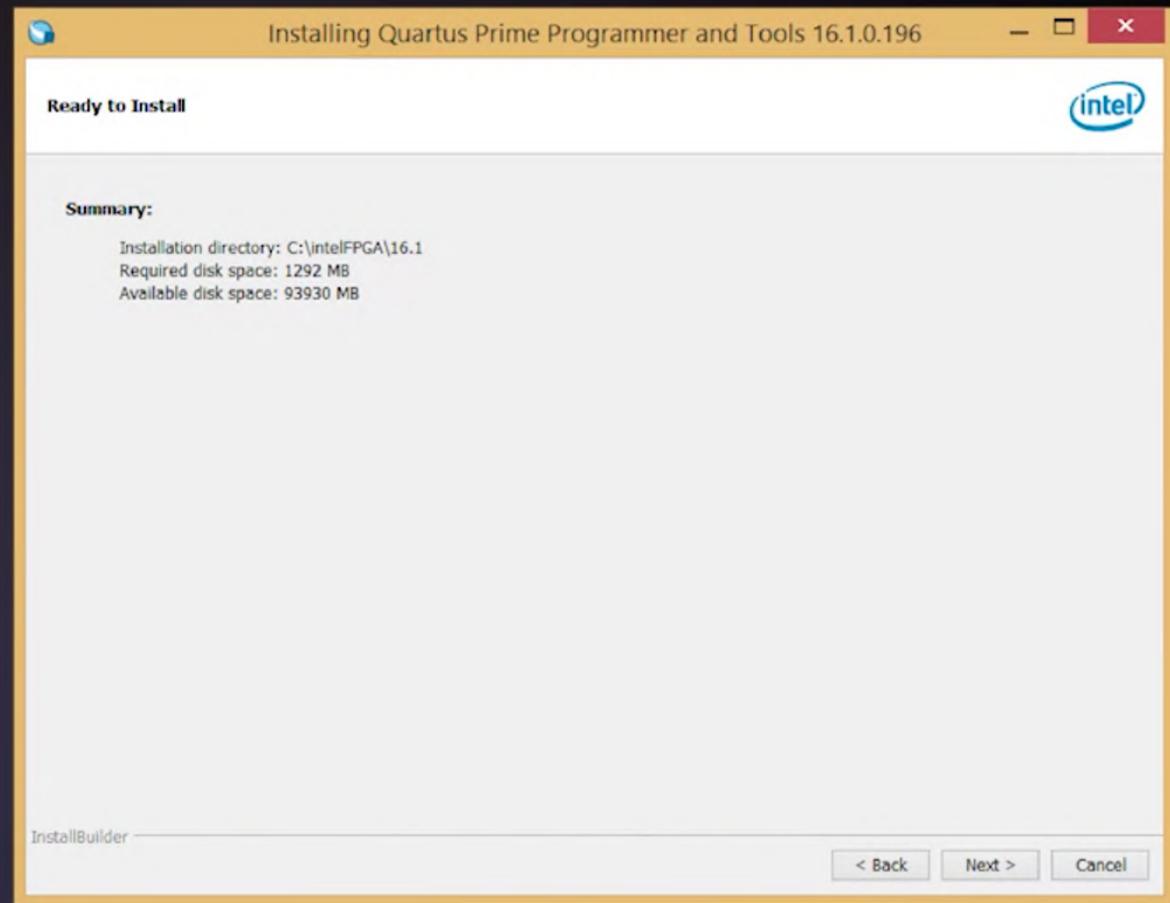


Choose a directory to install Quartus Prime Programmer.

The default is usually the safest.

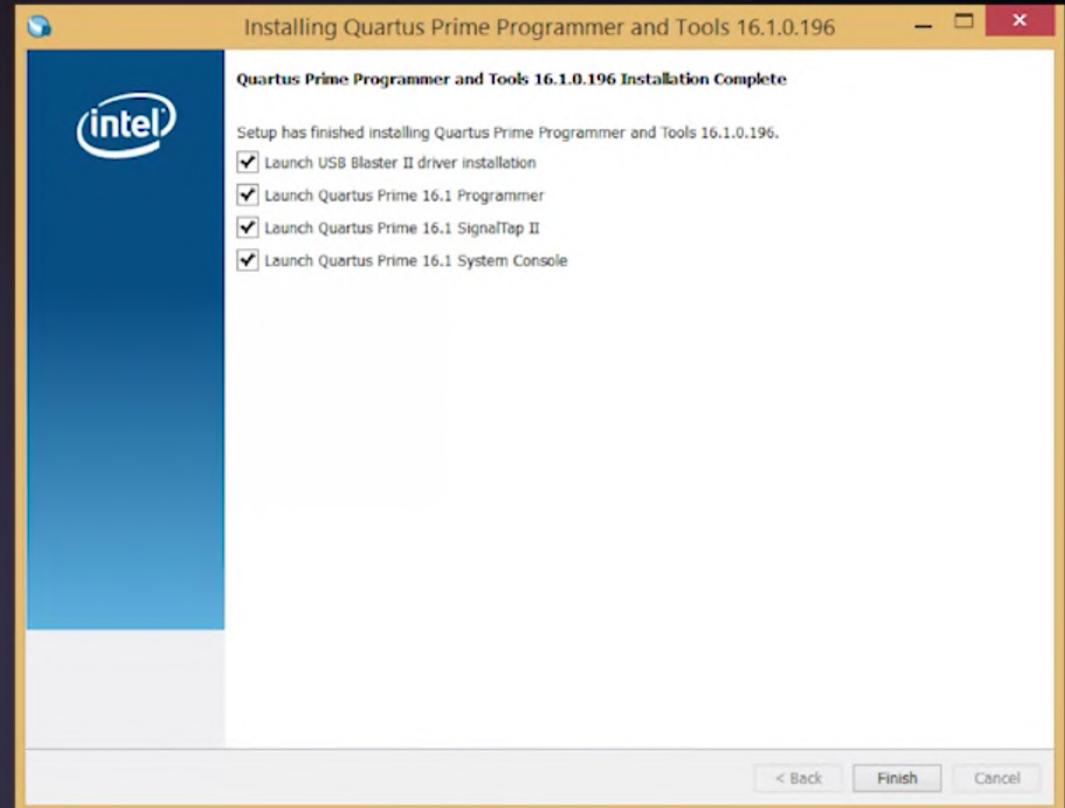


Now you are ready to install, provided you have enough memory.



You're done!

You can close any other programs that launch once you click finish.



More help is available at:

altera.com/support



Feel free to pause as
you enter commands.

Agenda for this Video



Create a project using the New Project Wizard.



Add files and libraries to the project.



Select a target device



Select EDA tools and settings.

In this video, you have learned:

- 1 How to create a project in Quartus Prime
- 2 How to select a target device for your project
- 3 How to add files and libraries to your project

In the next video, we will discover how to finish the design entry for this project.



Feel free to pause as
you enter commands.

Agenda for this Video

-  1 Create a schematic file for the top-level design.
-  2 Create a schematic symbol from an IP core multiplier.
-  3 Create a schematic symbol from an HDL file.
-  4 Wire the symbols together and to device pins.

Video 6 Summary

The various ways design entry can be performed for FPGAs

How to select and configure an IP core multiplier

How to create schematic symbols from HDL source files.

How to create a schematic of logic and IP blocks for FPGA implementation

Agenda for this Video

- 1 How to specify compiler settings to get the results you want
- 2 How to run full compilation
- 3 How to analyze compilation results.
- 4 How to create a design revision
- 5 How to archive a project

The Quartus Prime Compiler

A set of software modules that:

- Performs synthesis, fitting, assembly, and analysis
- Generates programming output files
- Runs modules together or independently
- Optimizes results through compiler settings
- Generates reports for analysis

Video 7 Summary

How to specify compiler settings to get the results you want

How to run full compilation

How to analyze compilation results.

How to create a design revision

How to archive a project

Agenda for this Video

- 1 View a design at the RTL level
- 2 View a design at the Technology level
- 3 Analyze a design using the Chip Planner
- 4 Analyze a design with the Power Play Power Analyzer

Video 8 Summary

How to view a design at the RTL level

How to view a design at the Technology level

How to analyze a design using the Chip Planner

How to determine an early power estimate of
an FPGA Design using the Power Play Analyzer

Agenda for this Video

- 1 Fundamentals of Timing Analysis
- 2 Timing Hazards and how to avoid them
- 3 Timing Analysis Terminology and Calculations
- 4 Using TimeQuest to do timing Analysis

Timing Analysis Fundamentals

Clocks are fundamental to modern digital electronics. They provide synchronization which is necessary for error-free data transfer.

As data on a digital line transitions from one value to another, during the transition time the data value will be incorrect. We need to wait until the data is stable before using it. The synchronizing signal is the clock.

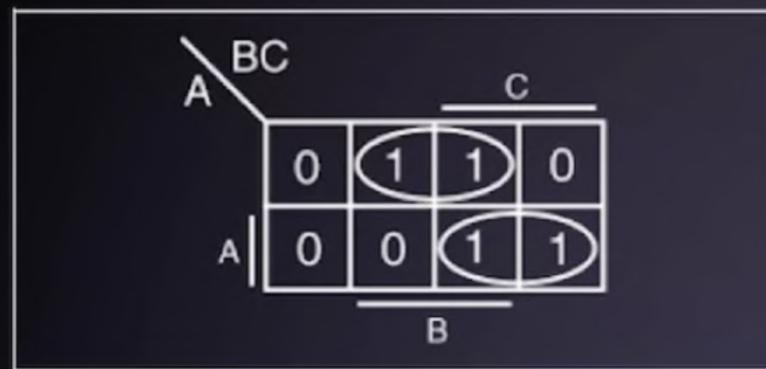
Timing Analysis Fundamentals

Data must be stable before and after the clock edge to be reliably transferred. If not properly synchronized, there will be a host of design issues like:

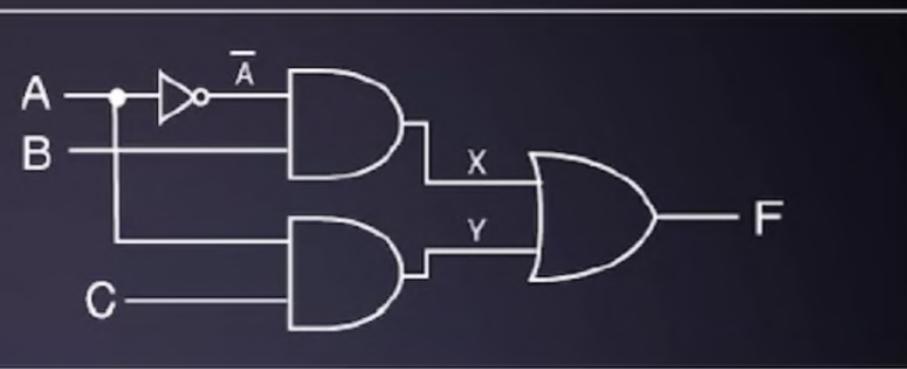
- Hazards
- Metastability
- Race Conditions
- Clock Skew

Timing Hazard

Static Hazards occur as the consequence of unequal delays in logic



(a)



(b)

Figure 21.6: The standard K-map approach to function reduction and the resulting circuit.

Timing Hazard

Static Hazards occur as the consequence of unequal delays in logic

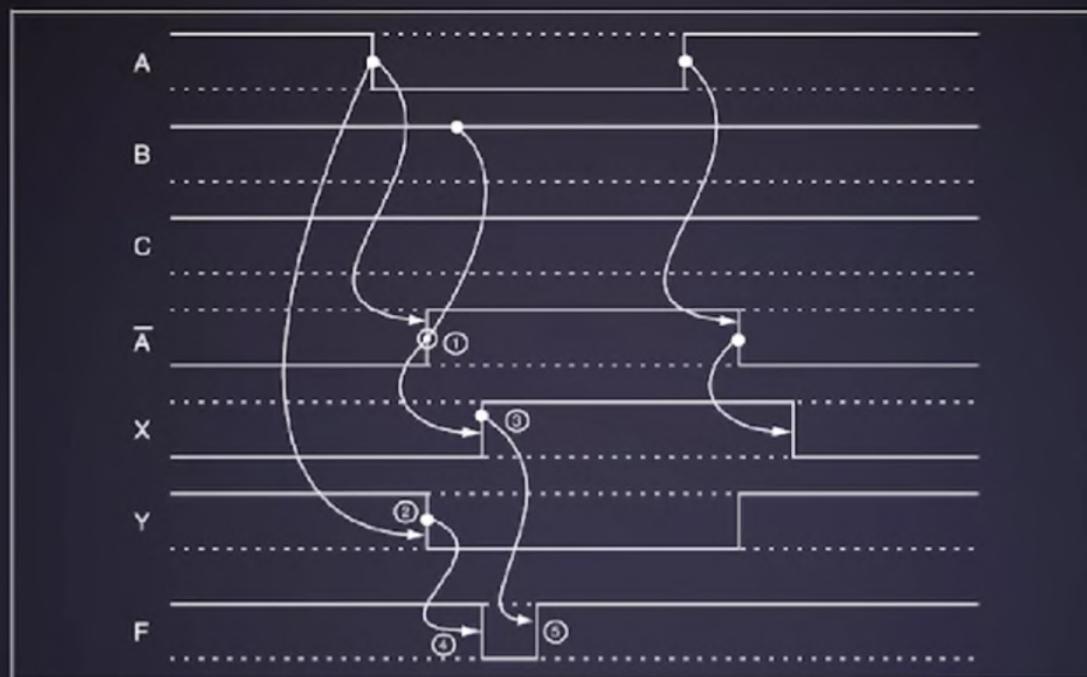
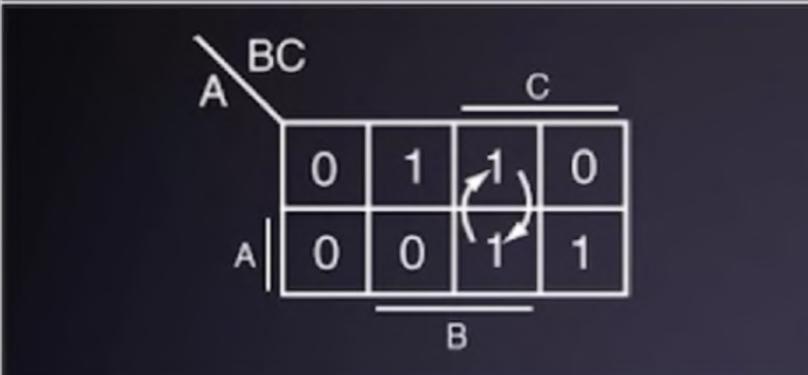


Figure 21.7: The timing diagram generated from the circuit of Figure 21.6.

Hazards - how to remove

One way to remove the hazard is through additional logic:

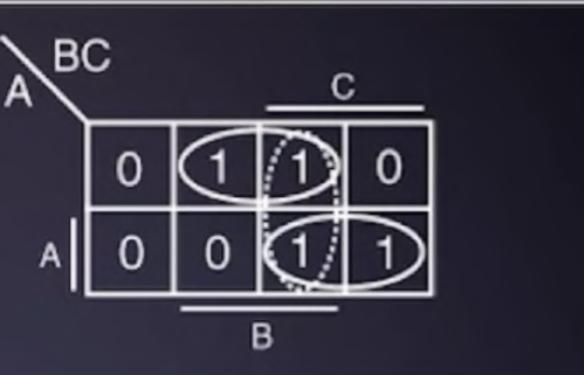


A Karnaugh map for three variables A, B, and C. The columns are labeled A (top-left), BC (top-right), and C (bottom-right). The rows are labeled A (left), B (middle), and A (right). The map shows the following values:

0	1	1	0
0	0	1	1
0	0	1	1

A dashed oval covers the cells (0,1) and (1,0), representing the cover term for the hazard at (1,1).

(a)



A Karnaugh map for three variables A, B, and C. The columns are labeled A (top-left), BC (top-right), and C (bottom-right). The rows are labeled A (left), B (middle), and A (right). The map shows the following values:

0	1	1	0
0	0	1	1
0	0	1	1

Two dashed ovals cover the cells (0,1), (1,0), and (1,1), representing the cover terms for the hazard at (1,1).

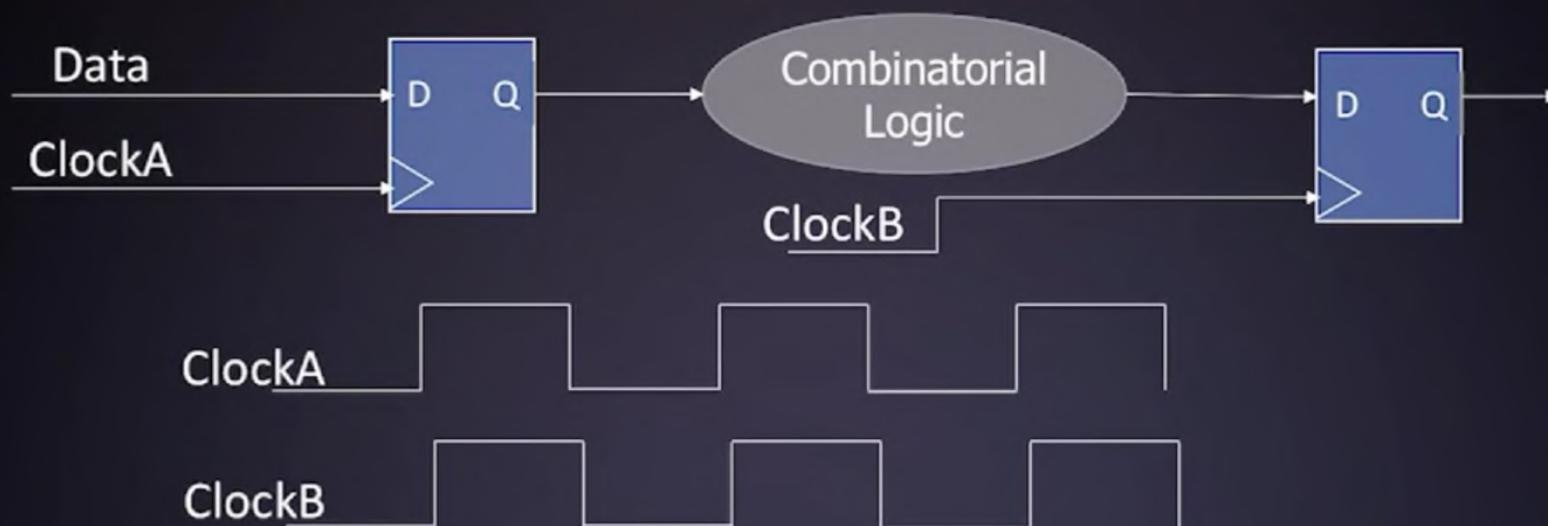
(b)

Figure 21.8: The transition of interest and the associated cover term for the function.

Another is using flip-flops and synchronous design.

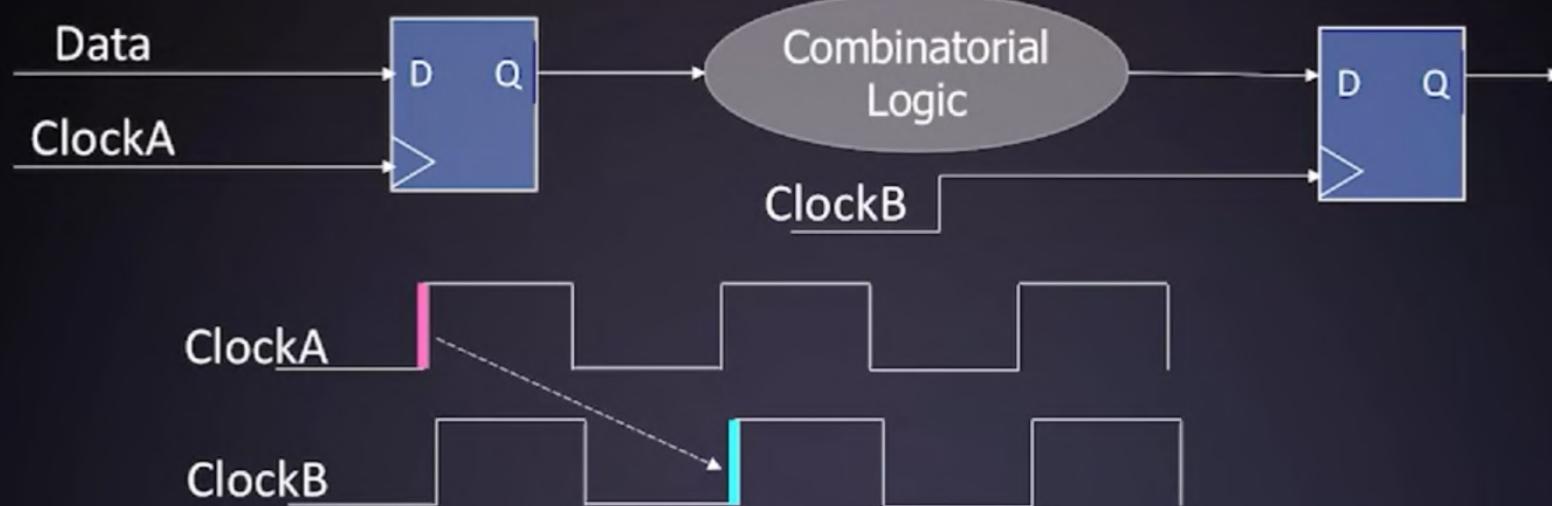
Timing Analysis Fundamentals

Fundamental synchronous circuit that static timing analyzers can analyze:



ClockA and ClockB may have a common source,
but the path delays may be different.

Timing Analysis Terminology



Launch Edge = Clock edge that activates the source register in a register-to-register path

Latch Edge = Clock edge that activates the destination register and captures the data

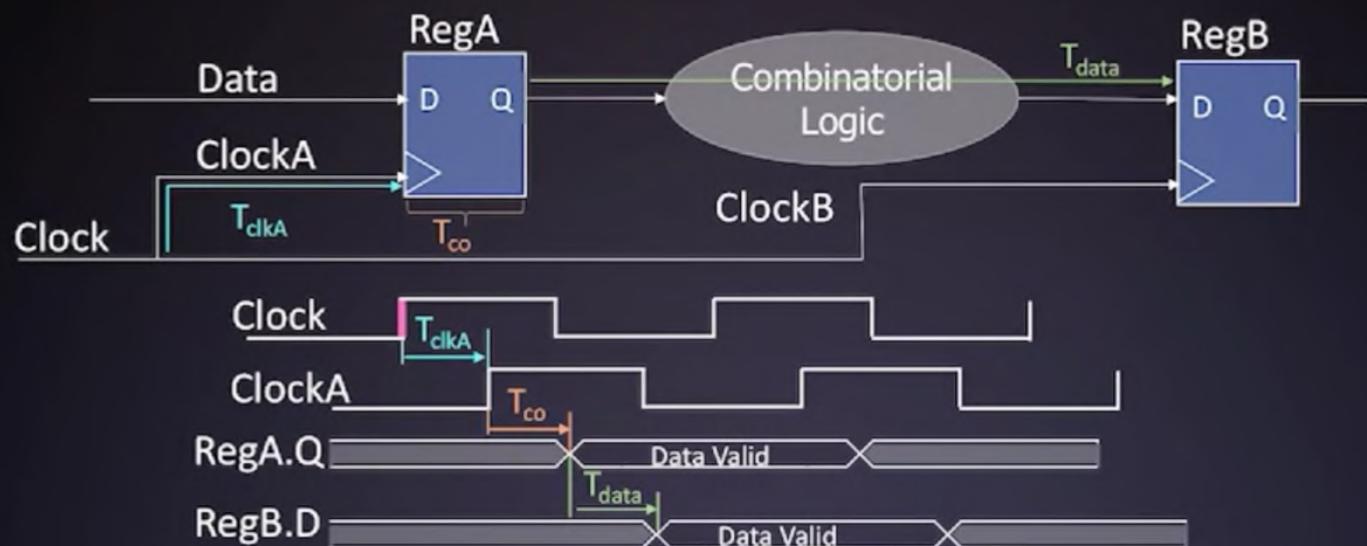
Timing Analysis Terminology



Setup Time = The minimum time the data signal must be stable BEFORE the clock edge

Hold Time = The minimum time the data signal must be stable after the clock edge

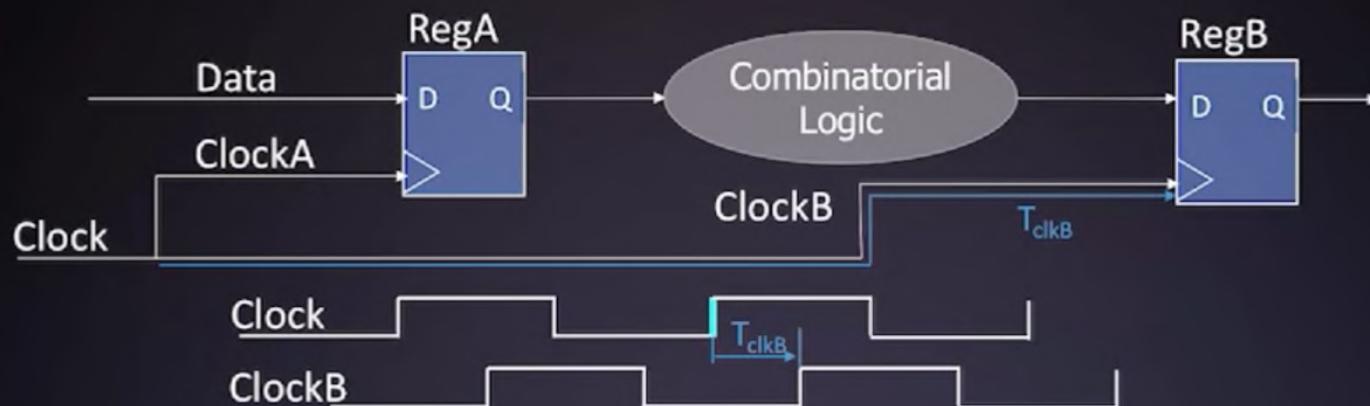
Timing Analysis Terminology



Data Arrival Time = the time for data to arrive at a destination register's D input from the common clock edge.

Data Arrival Time = Launch edge + T_{clkA} + T_{co} + T_{data}

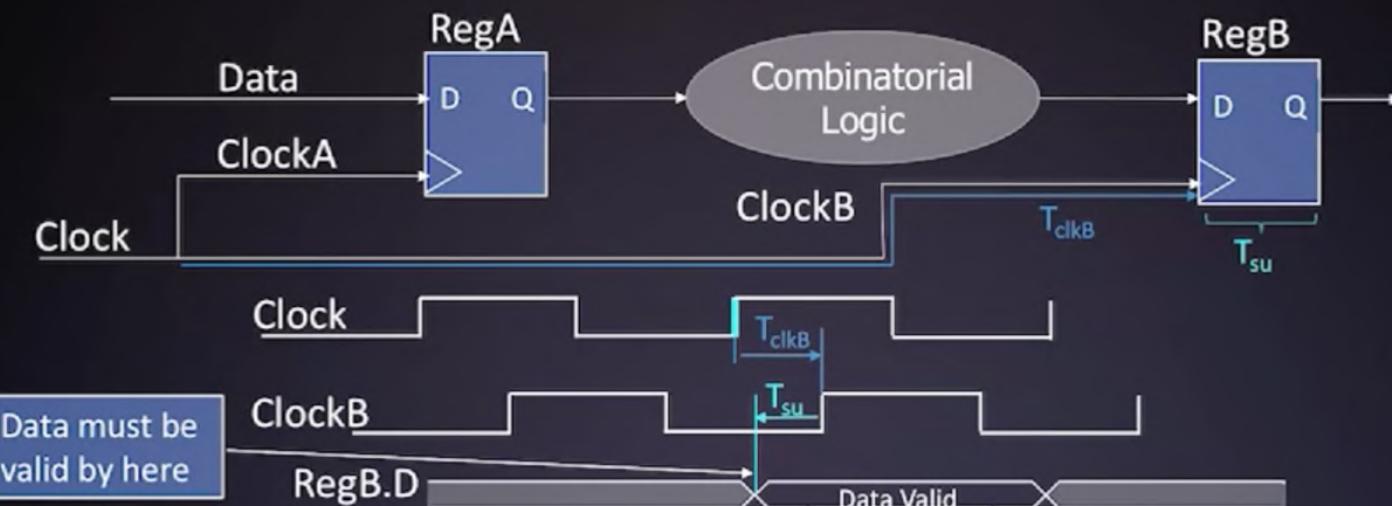
Timing Analysis Terminology



Clock Arrival Time = the time for clock to arrive at a destination register's clock input from the common clock edge.

Clock Arrival Time = Latch edge + T_{clkB}

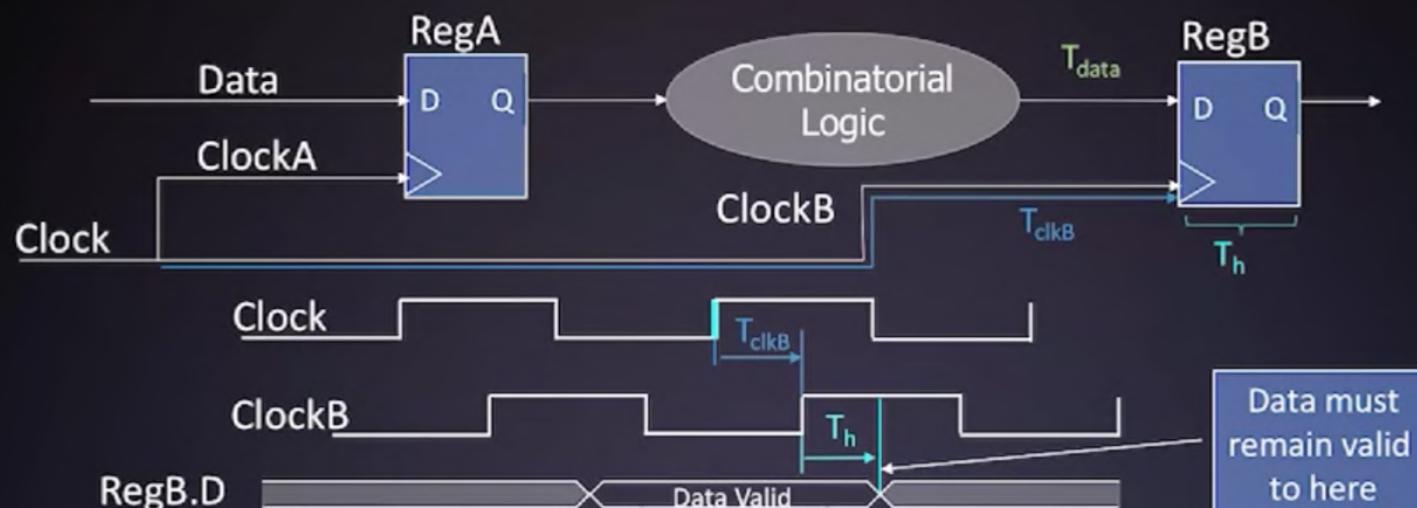
Timing Analysis Terminology



Data Required Time (Setup) = the minimum time required for the data to get latched into the destination register.

Data Required Time (Setup) = Clock Arrival Time – T_{su}

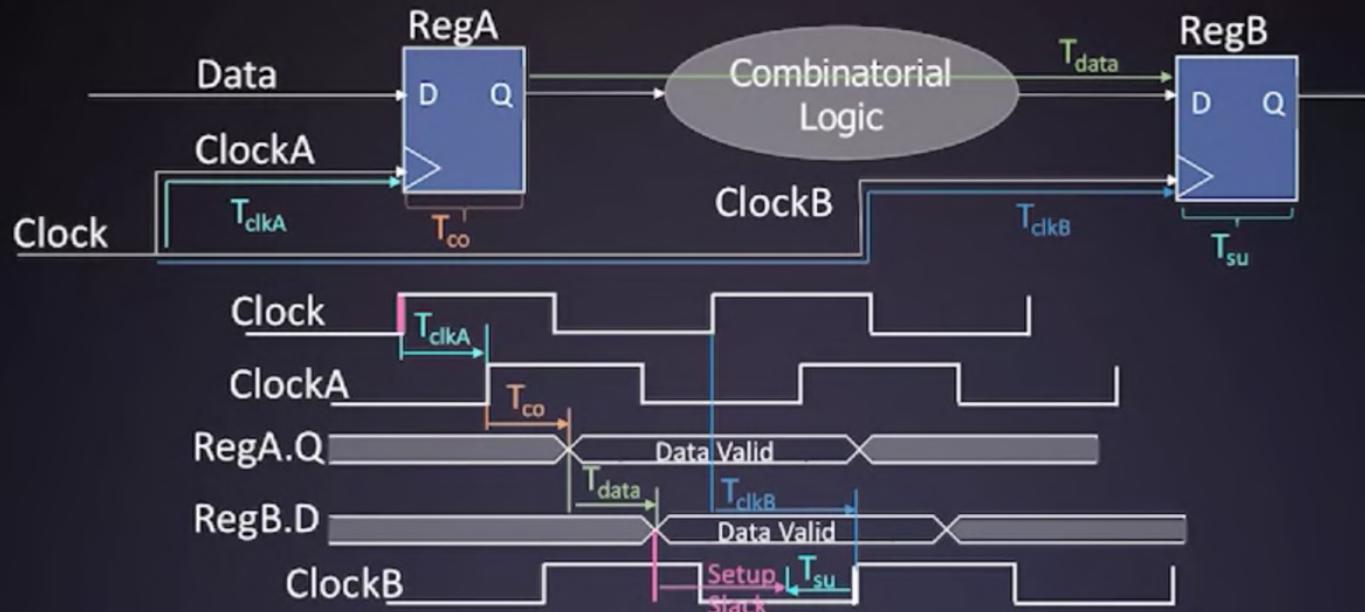
Timing Analysis Terminology



Data Required Time (Hold) = the minimum time required for the data to get latched into the destination register.

Data Required Time (Hold) = Clock Arrival Time + T_h

Timing Analysis Terminology

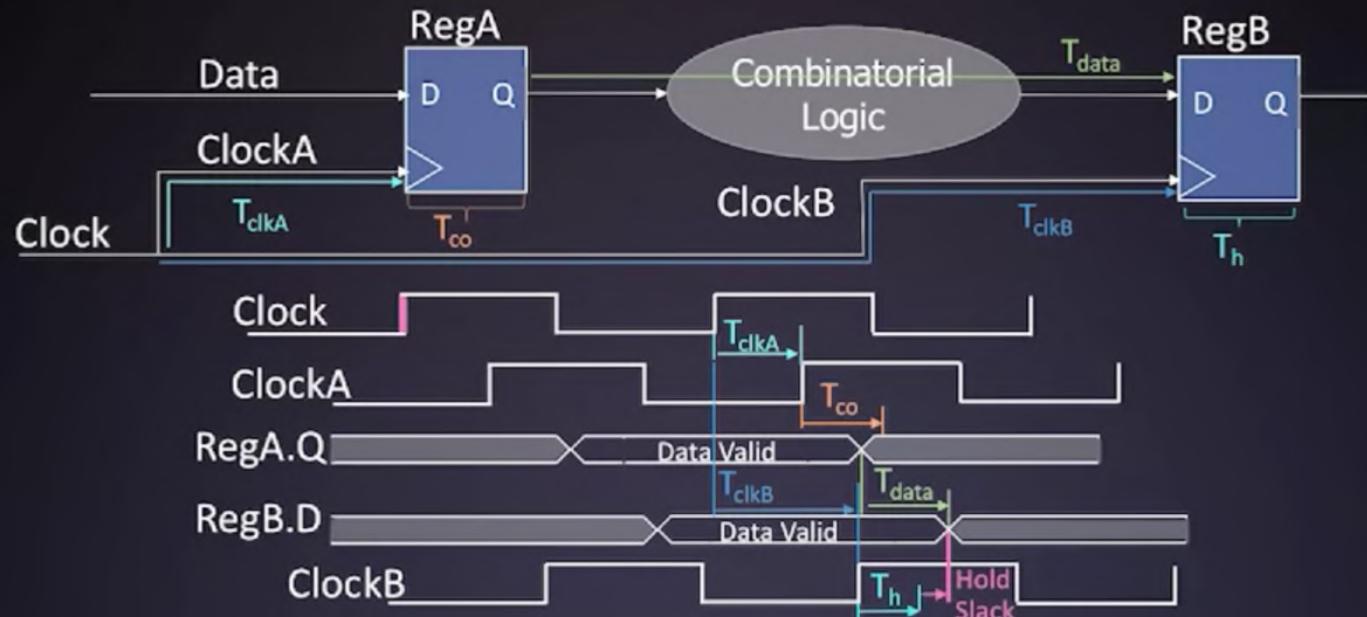


Setup Slack = The margin by which the setup timing requirement is met.

Setup Slack = Data Required Time – Data Arrival Time

Setup Slack = Clock Period + T_{clkB} – T_{su} - T_{clkA} - T_{co} - T_{data}

Timing Analysis Terminology

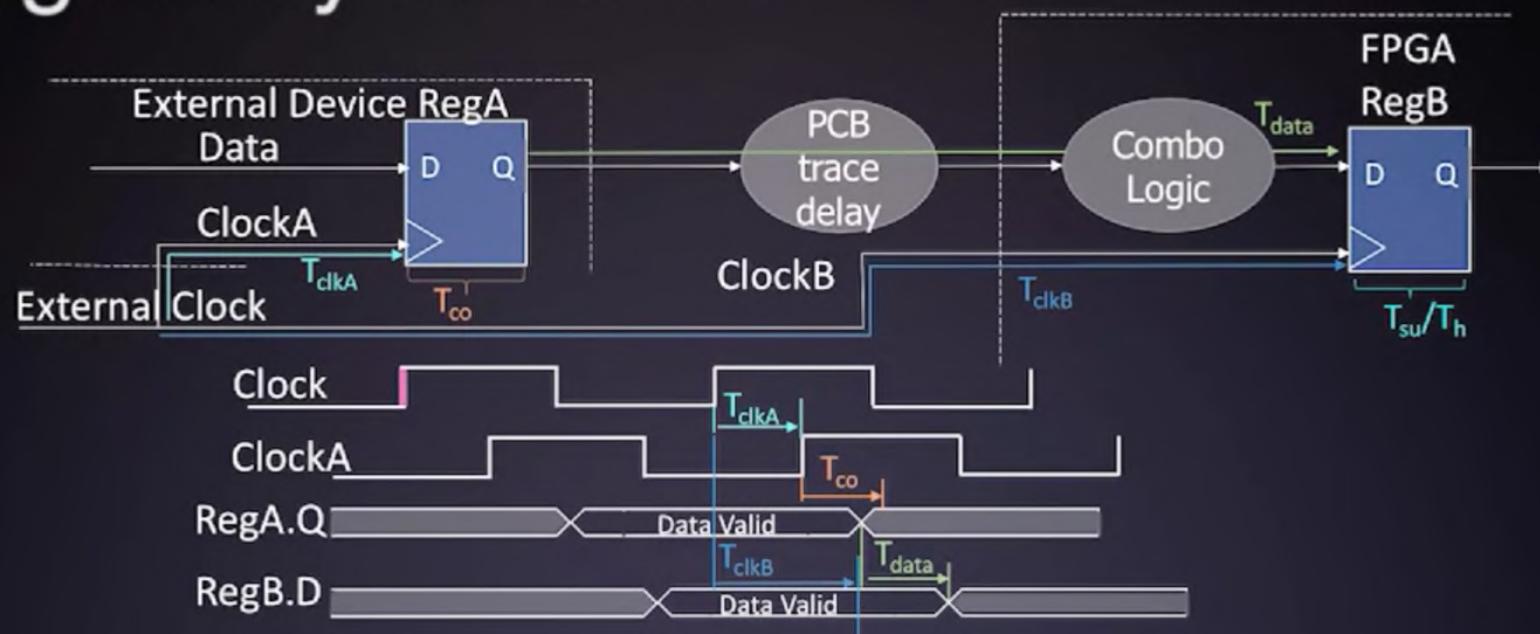


Hold Slack = The margin by which the hold timing requirement is met.

Hold Slack = Data Arrival Time – Data Required Time

Hold Slack = $T_{clkA} + T_{co} + T_{data} - T_{clkB} - T_h$

Timing Analysis for I/O



I/O timing analysis uses the same slack equations. This shows the input delay case.

The net time of arrival for inputs of the data relative to the clock is known as input delay.

The net delay time for outputs is known as output delay.

Why Know These Terms and Calculations?

Calculations are important when timing violations occur

- Need to be able to understand cause of violation

Example Causes

- Data path too long
- Requirements too short (incorrect analysis)
- Large clock skew signifying a gated clock, etc.

TimeQuest timing analyzer uses these

- Equations to calculate slack
- Terminology (launch and latch edges, Data Arrival Path, Data Required Path, etc.) in timing reports

Using TimeQuest for Analysis

Basic Steps to Using the TimeQuest Timing Analyzer

1. Generate timing netlist
2. Enter SDC constraints
 - a) Create and/or read in SDC file (recommended method)
 - b) Constrain design directly in console
3. Update timing netlist
4. Generate timing reports
5. Save timing constraints (optional)

Timing Constraints 101

User MUST enter constraints for all paths to fully analyze design

- Timing analyzer only performs slack analysis on constrained
- Constraints guide the fitter to place & route design in order to meet timing requirements
- Recommendation: Constrain all paths (at least clocks & I/O)

Not as difficult a task as it may sound

- Wildcards help to find signals and group them
- Single, generalized constraints cover many paths, even all paths in an entire clock domain

Timing Analysis Summary

Fundamentals of Timing Analysis

How Timing Hazards are caused and how to avoid them

Timing Analysis Terminology and Calculations

Guidelines for Using TimeQuest to do timing analysis

Agenda for this Video

- 1 Run the TimeQuest Timing Analyzer on an FGPA design
- 2 Create a Timing Netlist
- 3 Write Constraints in TimeQuest
- 4 Use the TimeQuest report to identify timing problems
- 5 Save a SDC constraints file
- 6 Achieve Timing Closure by timing driven compilation

TimeQuest Timing Netlist Terminology

Cell – basic device building blocks (LUTs, registers, multipliers, memory, PLLs)

Pin – input or output of a cell

Net – connections between pins

Port – top-level device pins

Video 10 Summary

How to Run the TimeQuest Timing Analyzer on an FPGA design

How to Create a Timing Netlist

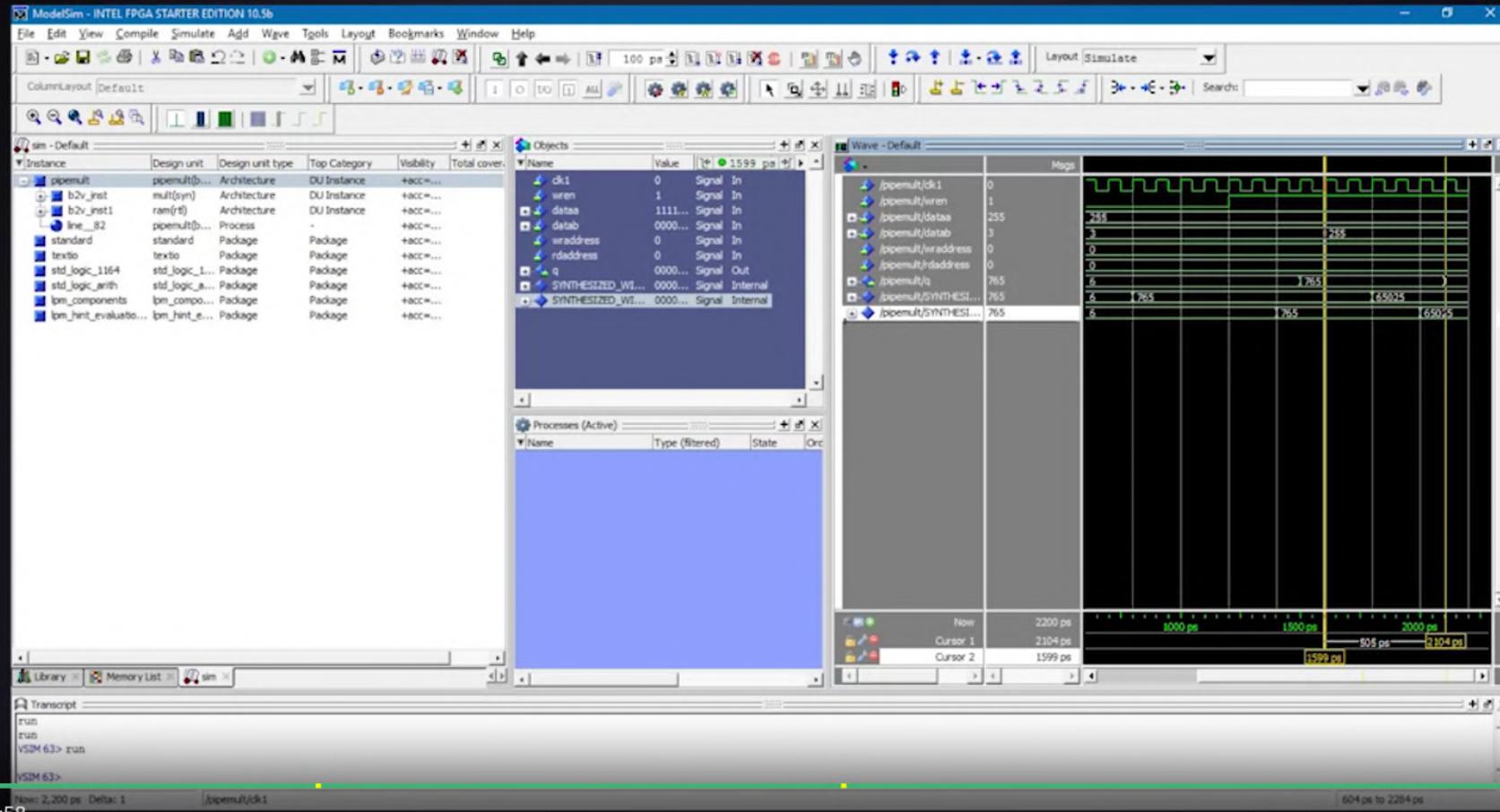
How to Write Constraints in TimeQuest

How to Achieve Timing Closure by timing driven compilation

Agenda for this Video

- 1 Convert a Schematic bdf file into a VHDL file for simulation
- 2 Start ModelSim from Quartus Prime
- 3 Add Signals to the Waveform Window
- 4 Create Stimulus to drive inputs in the simulation
- 5 Verify logic design by thoughtful design of test cases applied in the simulation.

Final Simulator Window



Video 11 Summary

How to Convert a Schematic bdf file into a VHDL file for simulation

How to start and configure ModelSim

How to Add Signals to the Waveform Window

How to Create Stimulus to drive inputs in the simulation

How to Verify logic design by thoughtful design of test cases applied in simulation.