



The Intel® Quartus® Prime Design Software: Foundation

Supplemental Files to Download



- Complete presentation in PDF format
- Lab exercise manual in PDF format
- Lab exercise files (executable ZIP file (**.exe**))
- All files contained in single **.zip** file
- Click link in email or go to **Resources** button to download (may need to hold ctrl to download)

Course Objectives

- Create a new Intel® Quartus® Prime Design Software project
- Choose supported design entry methods
- Compile a design into a programmable logic device (PLD)
- Locate resulting compilation information
- Create design constraints (assignments & settings)
- Manage I/O assignments
- Prepare for programming/configuring a PLD

Class Agenda

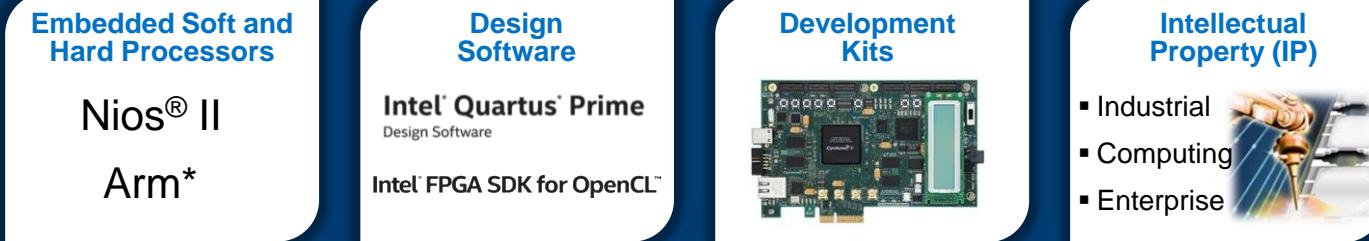
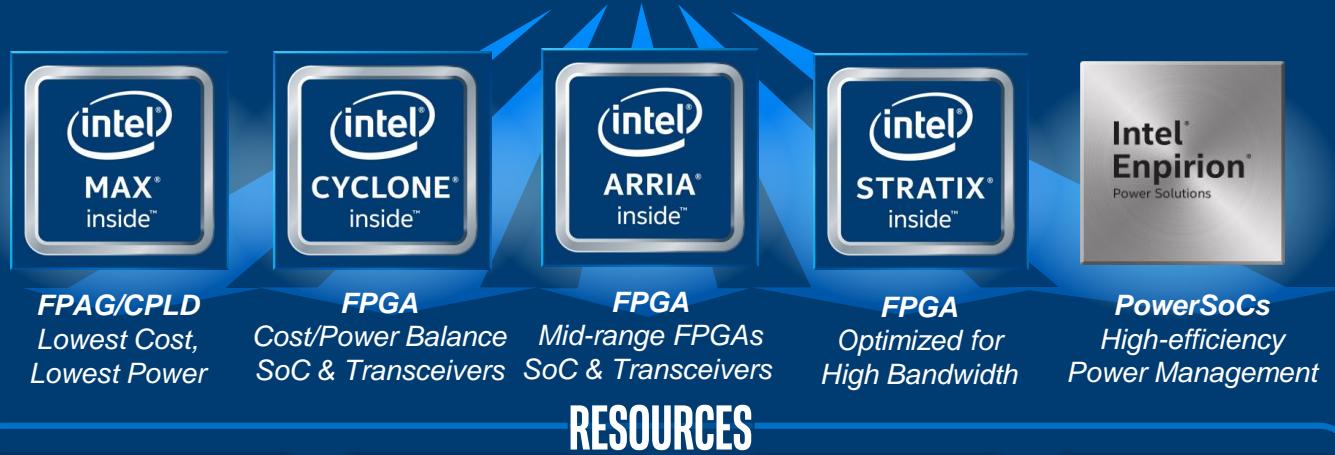
- Intro to Intel® FPGA devices, and Tools
- Intel® Quartus® Prime Feature Overview
- Project and IP Management
 - Exercise 1
- Design Methodology
- Design Entry
 - Exercise 2
- Compilation
 - Exercise 3
- Settings & Assignments
 - Exercise 4
- I/O Planning
- Programming/Configuration
 - Exercise 5



The Intel® Quartus® Prime Design Software: Foundation

Introduction to Intel® FPGA, Devices, and Tools

Innovation Across the Board



Intel® Quartus® Prime Software – Three Editions



Intel® Quartus® Prime
Design Software

Web Edition



Lite Edition (LE)

Subscription

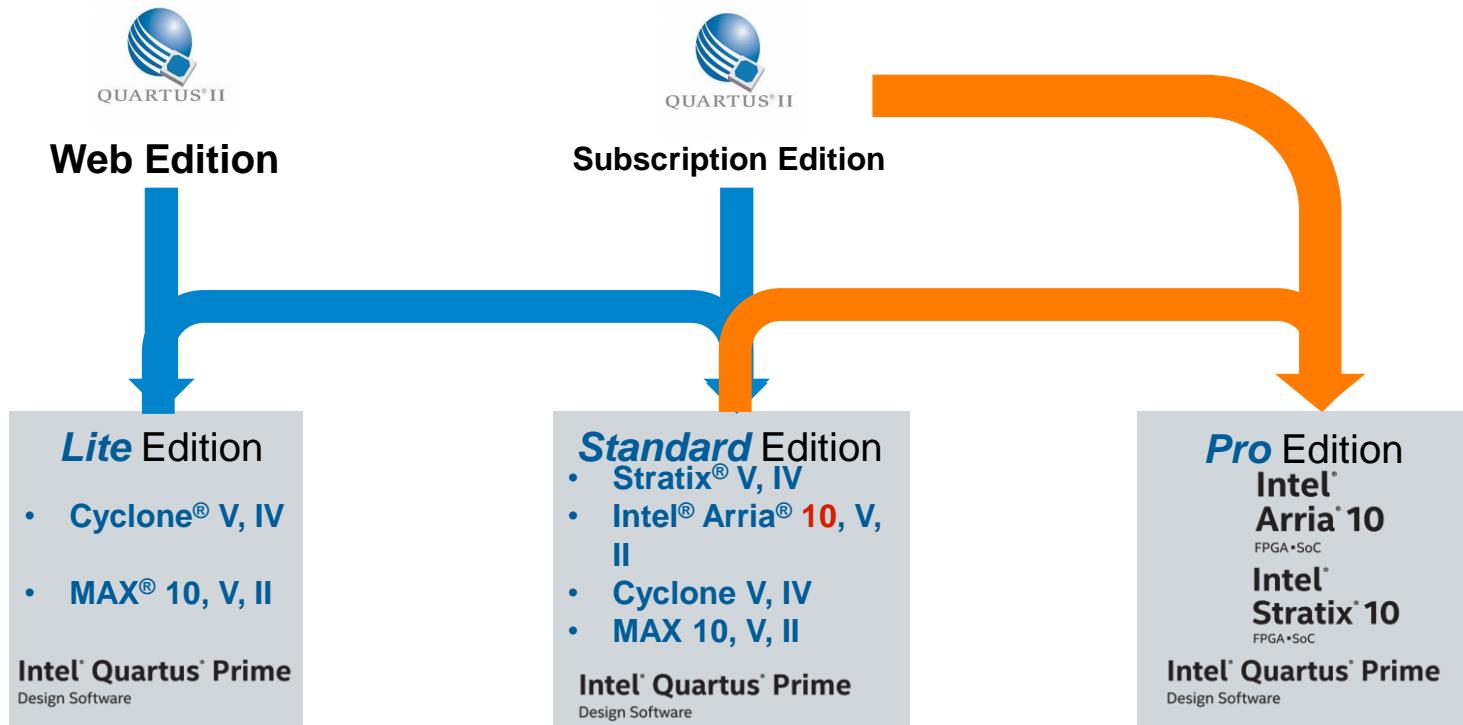


Standard Edition (SE)

Pro Edition (PE)

[Feature Comparison available on the Intel® FPGA web site](#)

Migration Paths



Free online training: [Migrating to the Intel® Quartus® Prime Pro Edition Software](#)

Intel® Quartus® Prime Software Licensing

Web Edition → Lite Edition (LE)

- No change
- No license file required

Subscription Edition → Standard Edition (SE)

- No change
- Previous Subscription Edition license will work with Standard Edition

Pro Edition (PE)

- Requires a new feature line



The Intel® Quartus® Prime Design Software: Foundation

Intel® Quartus® Prime Design Software Feature Overview

Intel® Quartus® Prime Design Software

- Fully-integrated development tool
 - Multiple design entry methods
 - Logic synthesis
 - Place & route
 - Device programming
- Simulation
 - Supports standard HDL simulation tools
 - Includes ModelSim*-Intel® FPGA Starter Edition tool
 - Optional upgrade to ModelSim-Intel FPGA Edition tool
 - See comparison
 - <https://www.altera.com/products/design-software/model---simulation/modelsim-altera-software.html>

Intel® Quartus® Prime Design Software Features

Intel® Quartus® Prime Design Software Tool/Feature	
Operating system	<ul style="list-style-type: none">▪ 64-bit Windows and Linux support
Licensing	<ul style="list-style-type: none">▪ Node-locked and network licensing support
Project creation	<ul style="list-style-type: none">▪ New Project Wizard
Design entry	<ul style="list-style-type: none">▪ Text Editor (HDL support)▪ Schematic Editor▪ State Machine Editor▪ IP Catalog & IP Parameter Editor▪ Platform Designer or Platform Designer system design tool▪ DSP Builder for Intel® FPGA Standard/Advanced Blockset▪ OpenCL® platform support▪ 3rd-party design entry tool support
Constraint (assignment) entry	<ul style="list-style-type: none">▪ Assignment Editor▪ Text Editor support of Synopsys Design Constraints (SDC)▪ Pin Planner▪ Interface Planner (Pro Edition only)▪ Scripting (Tcl) support
Design processing/compilation (synthesis and fitting)	<ul style="list-style-type: none">▪ Intel Quartus Integrated Synthesis software (QIS)▪ 3rd party EDA synthesis tool support▪ Intel Quartus Prime Software Fitter (<i>Lite & Standard</i>)
Design evaluation and debugging	<ul style="list-style-type: none">▪ RTL Viewer▪ Technology Map Viewers▪ State Machine Viewer

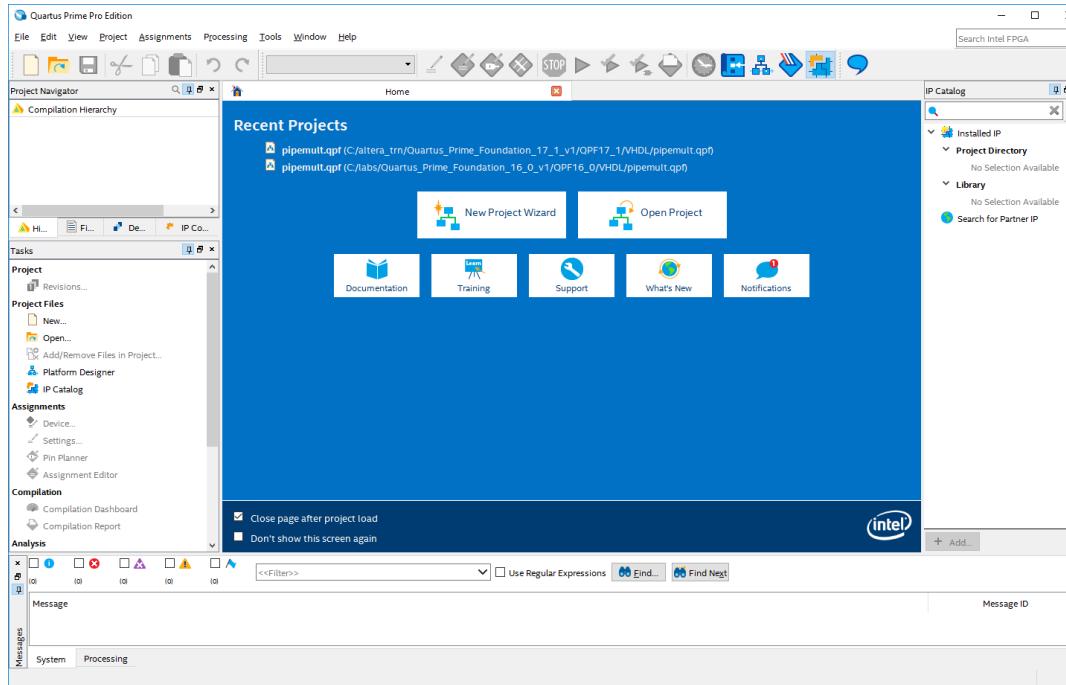
Intel® Quartus® Prime Design Software Features

Category	Intel® Quartus® Prime Design Software Tool/Feature		
Power analysis	<ul style="list-style-type: none">▪ Power Analyzer		
Static timing analysis	<ul style="list-style-type: none">▪ Timing Analyzer		
Simulation	<ul style="list-style-type: none">▪ ModelSim*-Intel® FPGA Starter Edition▪ ModelSim-Intel FPGA Edition▪ 3rd-party EDA simulation tool support		
Chip layout viewing and modification	<ul style="list-style-type: none">▪ Chip Planner▪ Resource Property Editor		
Programming file generation	<ul style="list-style-type: none">▪ Intel Quartus Prime Assembler software		
FPGA/CPLD programming	<ul style="list-style-type: none">▪ Intel Quartus Prime Programmer software		
Hardware debugging tools	<ul style="list-style-type: none">▪ Signal Tap embedded logic analyzer▪ In-System Sources and Probes▪ Signal Probe incremental routing	<ul style="list-style-type: none">▪ System Console▪ Transceiver Toolkit▪ In-System Memory Content Editor	
Design optimization and productivity improvement	<ul style="list-style-type: none">▪ Rapid Recompile▪ Intel Quartus Prime incremental compilation (<i>Standard Edition only</i>)▪ Incremental Optimization (<i>Pro Edition only</i>)▪ Physical synthesis optimization▪ Design Space Explorer II (DSE)		

Intel® Quartus® Prime Design Software Edition Use in this Training

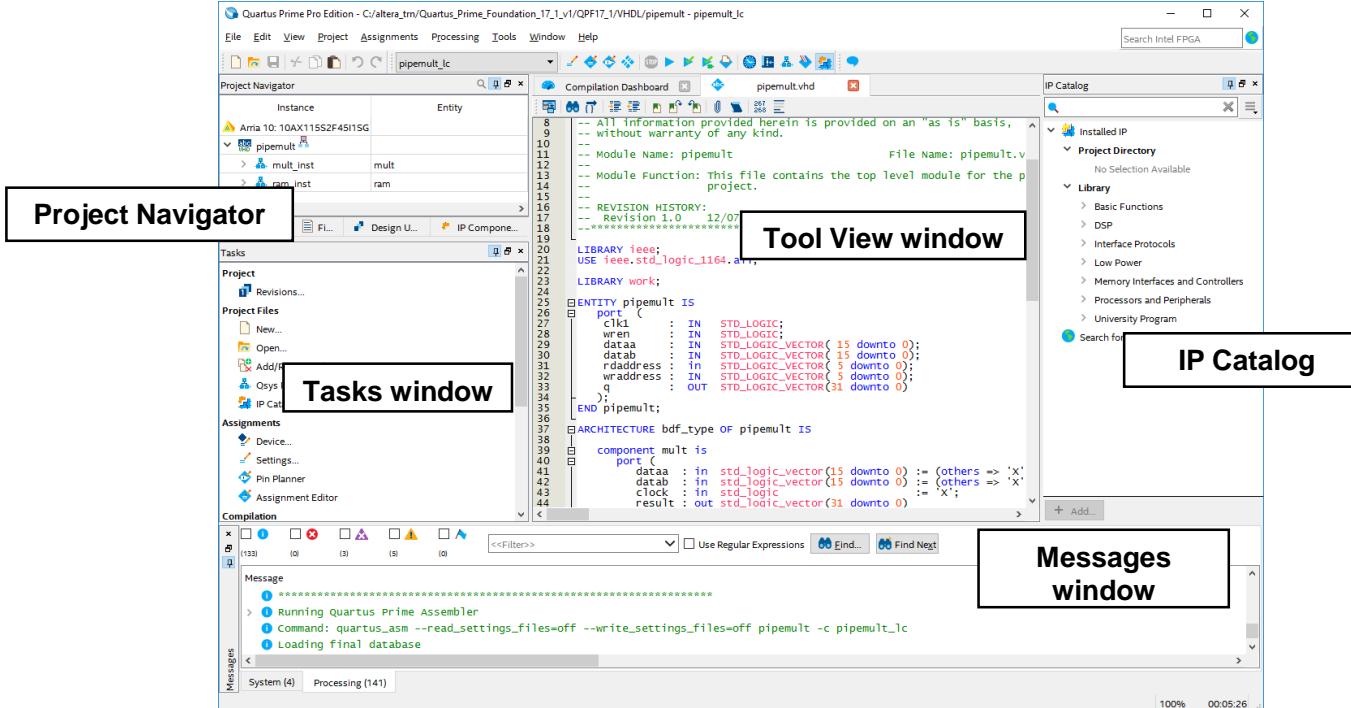
- Training focuses on Pro Edition
 - Tool usage
 - Screen shots
 - Lab exercises
- Most features identical between all 3 editions
- Pro Edition-specific features will be highlighted
- Standard Edition version (17.1) of this training, including labs
 - <https://www.altera.com/support/training/course.html?courseCode=ODSW1110>

Welcome to the Intel® Quartus® Prime Design Software!

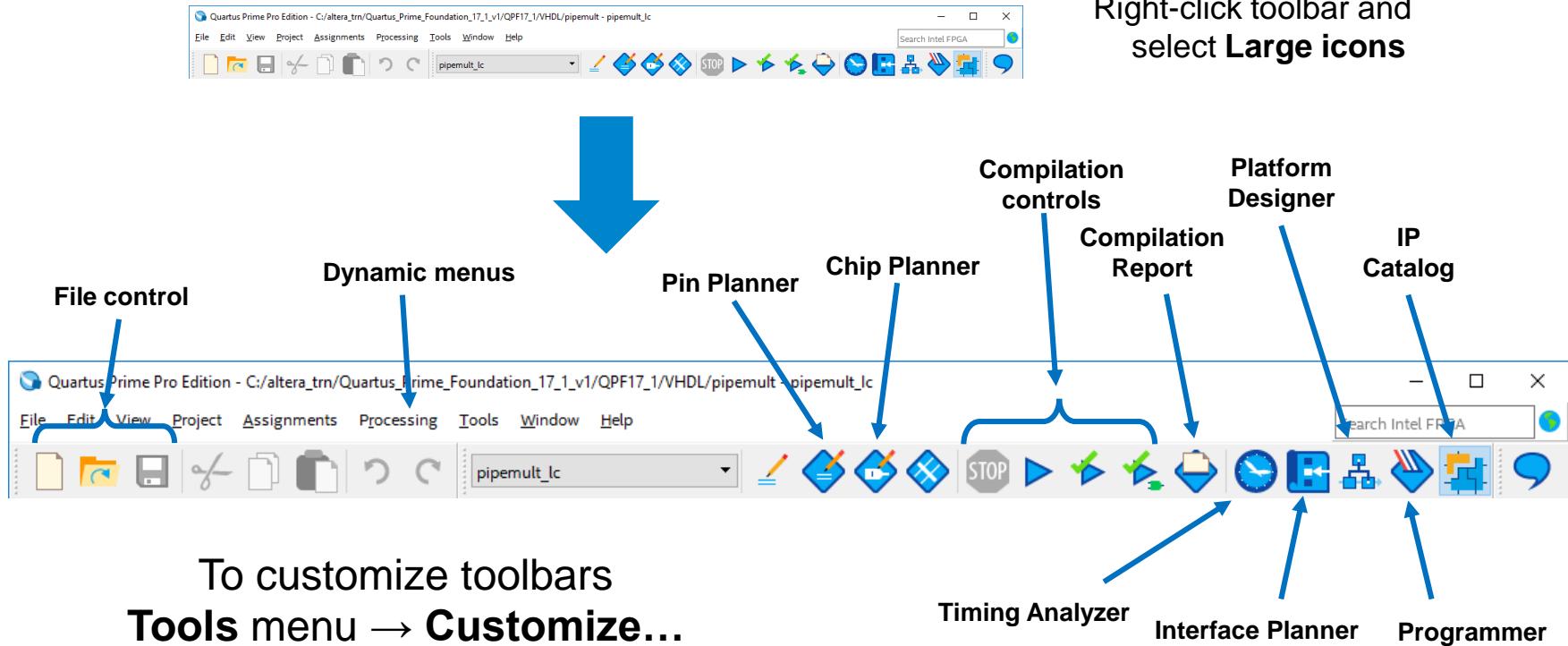


Turn on or off in
Tools menu → **Options**

Intel® Quartus® Prime Design Software Default Operating Environment

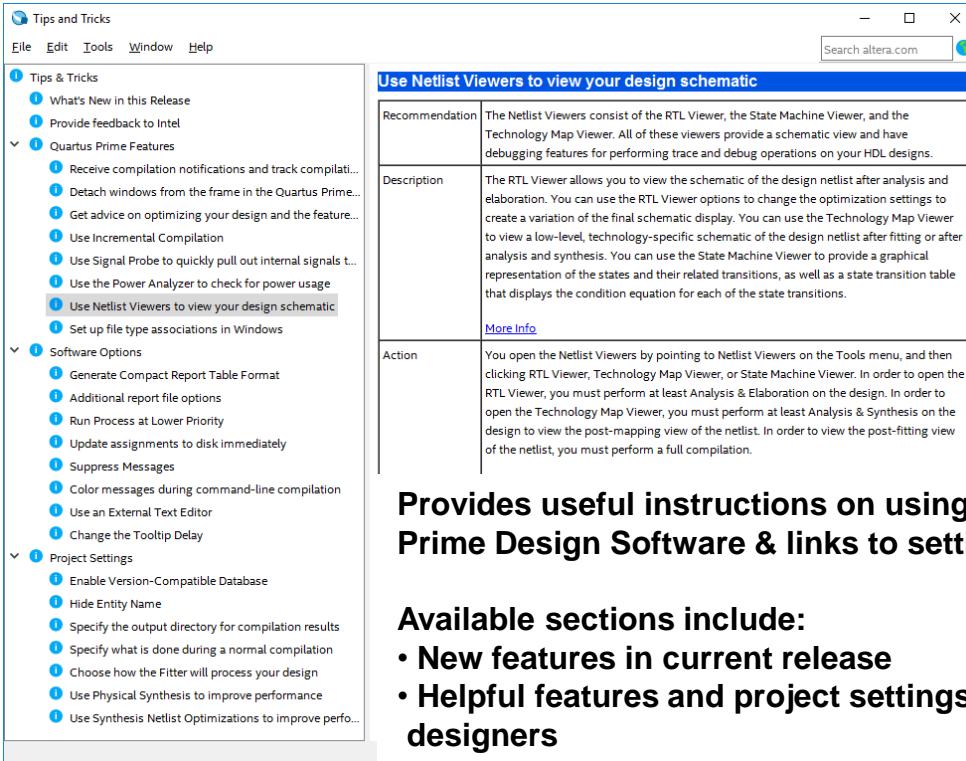


Main Toolbar



Tips & Tricks Advisor

Help menu → Tips & Tricks



The screenshot shows the 'Help' menu of the Intel Quartus Prime software. The 'Tips & Tricks' section is expanded, displaying various tips categorized under 'Quartus Prime Features', 'Software Options', and 'Project Settings'. One tip, 'Use Netlist Viewers to view your design schematic', is highlighted and expanded to show detailed instructions and a 'More Info' link.

Use Netlist Viewers to view your design schematic

Recommendation	The Netlist Viewers consist of the RTL Viewer, the State Machine Viewer, and the Technology Map Viewer. All of these viewers provide a schematic view and have debugging features for performing trace and debug operations on your HDL designs.
Description	The RTL Viewer allows you to view the schematic of the design netlist after analysis and elaboration. You can use the RTL Viewer options to change the optimization settings to create a variation of the final schematic display. You can use the Technology Map Viewer to view a low-level, technology-specific schematic of the design netlist after fitting or after analysis and synthesis. You can use the State Machine Viewer to provide a graphical representation of the states and their related transitions, as well as a state transition table that displays the condition equation for each of the state transitions.
Action	You open the Netlist Viewers by pointing to Netlist Viewers on the Tools menu, and then clicking RTL Viewer, Technology Map Viewer, or State Machine Viewer. In order to open the RTL Viewer, you must perform at least Analysis & Elaboration on the design. In order to open the Technology Map Viewer, you must perform at least Analysis & Synthesis on the design to view the post-mapping view of the netlist. In order to view the post-fitting view of the netlist, you must perform a full compilation.

Provides useful instructions on using the Intel® Quartus® Prime Design Software & links to settings

Available sections include:

- **New features in current release**
- **Helpful features and project settings available to designers**

Built-In Help System

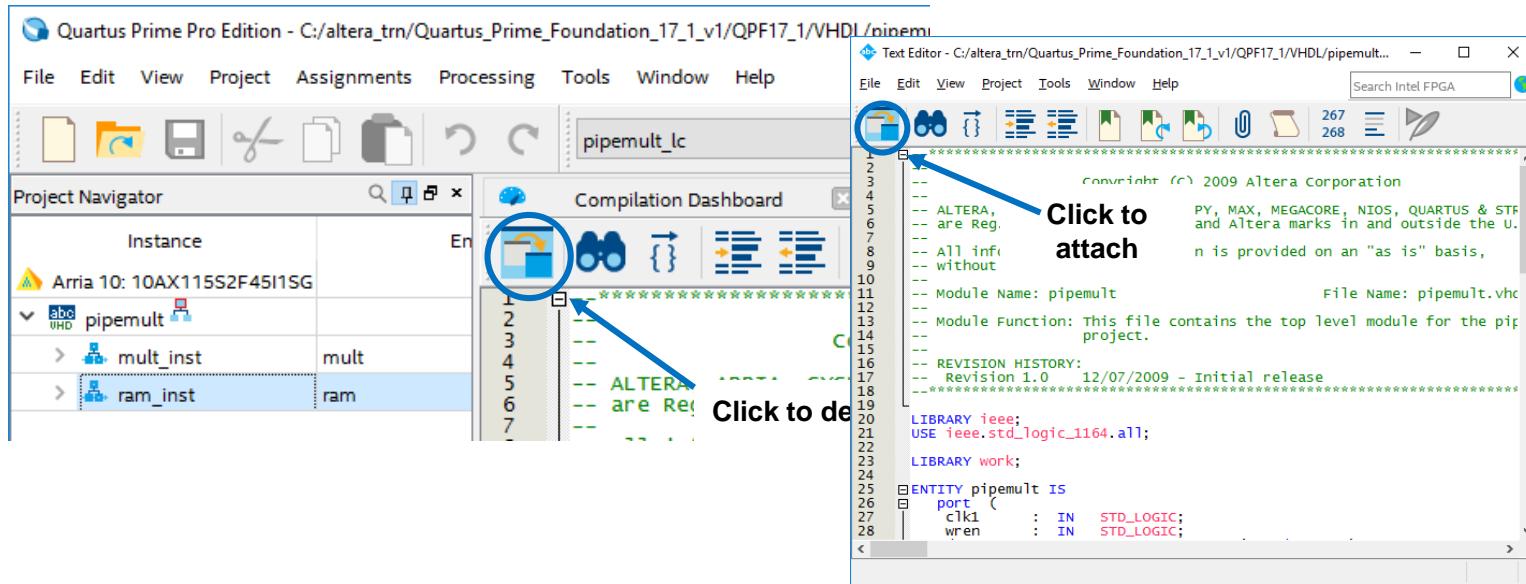
The image shows two screenshots of the Quartus Prime Help system. The left screenshot shows the built-in help interface with a sidebar containing links like 'Help Topics', 'Message List', 'Devices and Adapters', etc. The right screenshot shows a web browser displaying the Quartus Prime Help version 17.0, which includes a search bar and a sidebar with links to various help topics.

Note: Bug in 17.1 causes issues with Help in Chrome. Workaround is to click **OK in JavaScript dialog that appears, then change URL to [index_frames.htm](#) (instead of .html).**

Web browser-based help allows for easy search in page

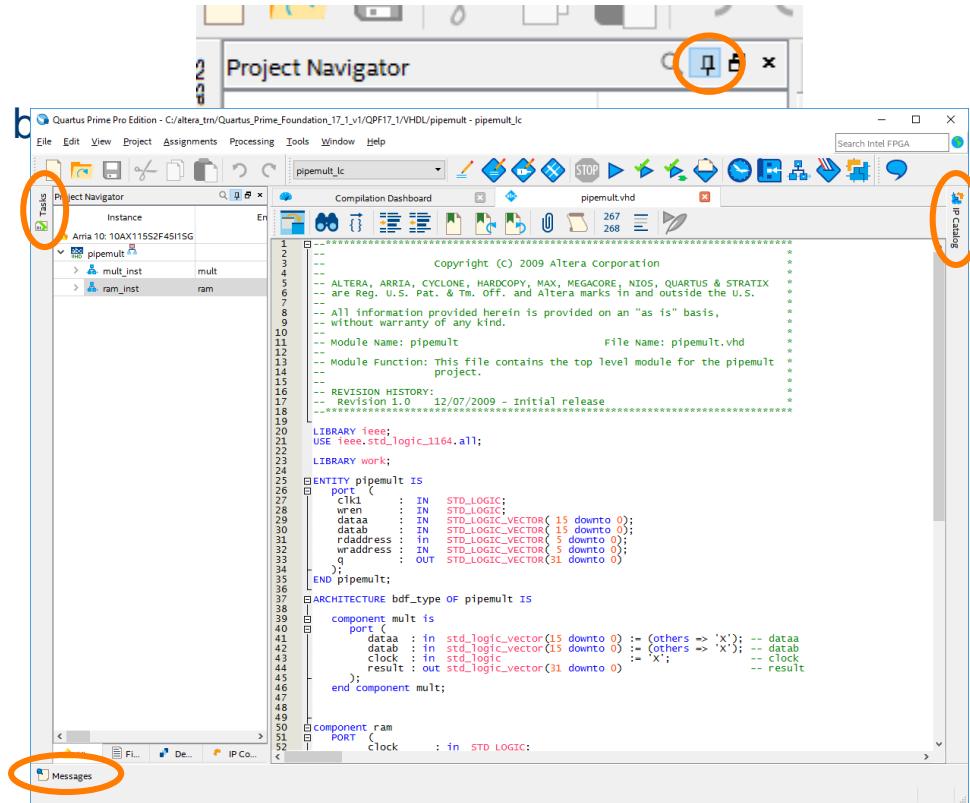
Detachable Windows

- Separate child windows from the Intel® Quartus® Prime Design Software GUI frame (Window menu → Detach/Attach Window)



Auto Hide/Show Dockable Windows

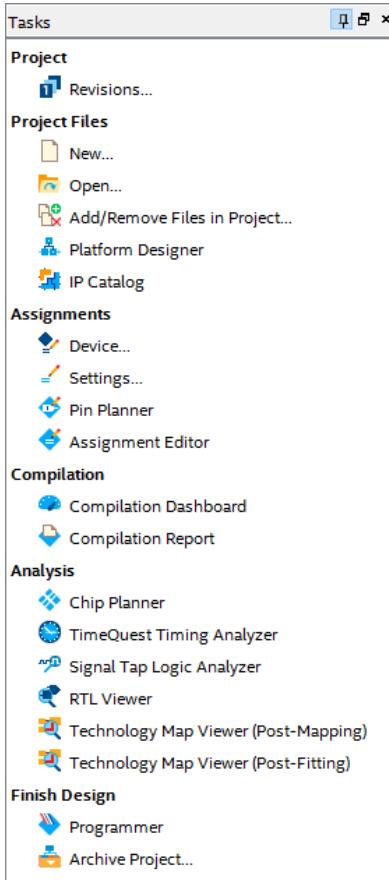
- Quickly hide/unhide dockable windows by “unpinning”
- Hover over a window’s tab to make it temporarily visible
- Re-pin to keep the window visible



Tasks Window

- Easy access to most Intel® Quartus® Prime Design Software functions
- Organized into stages of design flow

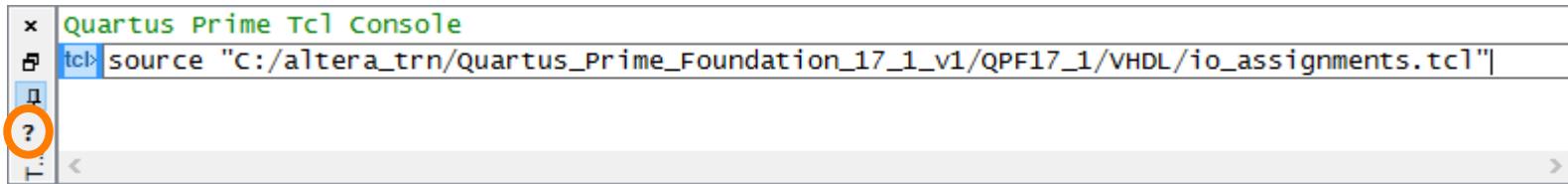
Click any task to execute



Tcl Console Window

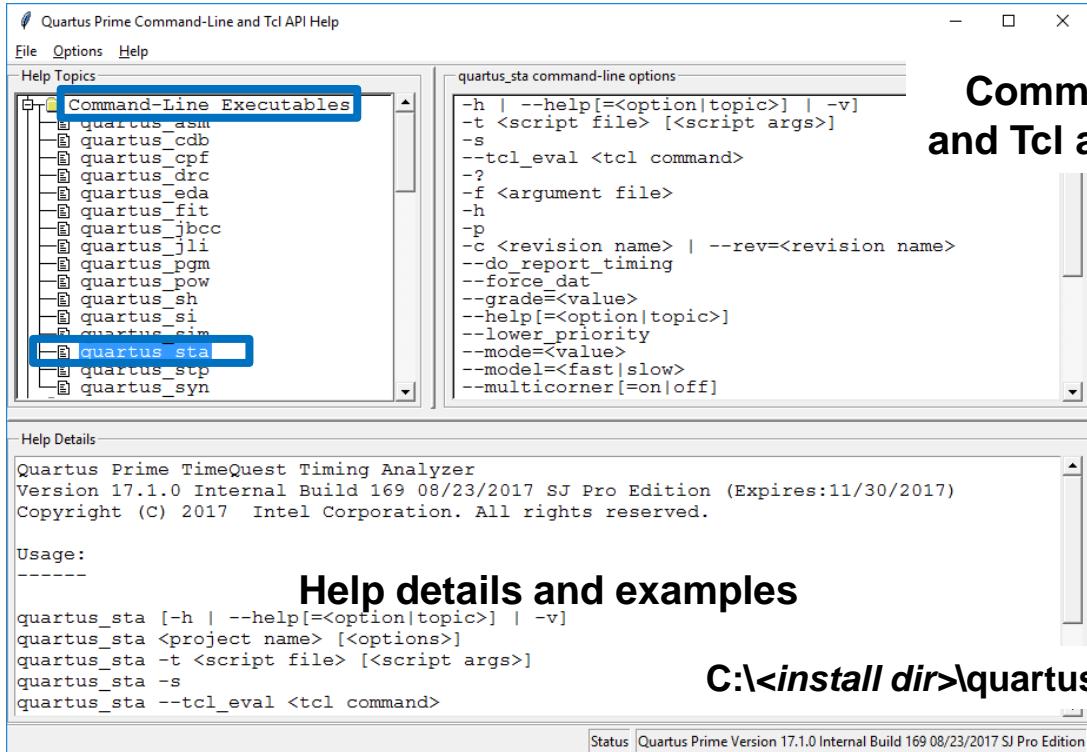
- Enter and execute Tcl commands directly in the GUI

View menu → Tcl Console

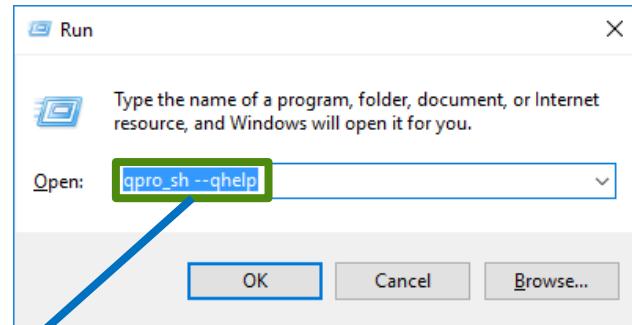


- Execute from command-line using Tcl shell
 - `qpro_sh -s`
- Run complete scripts from **Tools** menu → **Tcl Scripts...**

Tcl and Command Line Help



Command-line and Tcl arguments



Further Information

- Intel® Quartus® Prime Design Software Handbook always available online in pdf format
- <http://www.altera.com>: The diagram shows a navigation path from the Altera website. It starts with a blue button labeled "MENU ≡". An arrow points to a blue button labeled "SUPPORT". Another arrow points to a white box titled "Support Resources" containing four links: "Documentation" (highlighted in blue), "Knowledge Base", "Communities", and "Design Examples". A final arrow points to a white box titled "Design Tools, IP and Design Examples" containing icons for "Design Tools" (wrench and gear) and "IP & Design Examples" (lightbulb).
Intel® Quartus® Prime Software
- https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/qts/qts-qpp-handbook.pdf
- Intel® Quartus® Prime Software Handbook Volume 2 chapters
 - Command-Line Scripting
 - Tcl Scripting

Further Information

Free on-line training classes

- Introduction to Tcl
 - <http://www.altera.com/education/training/courses/ODSW1180>
- Intel® Quartus® Prime Design Software Tcl Scripting
 - <http://www.altera.com/education/training/courses/ODSW1190>

Tcl references on-line

- Good reference: <http://tmml.sourceforge.net/doc/tcl/>



The Intel® Quartus® Prime Design Software: Foundation

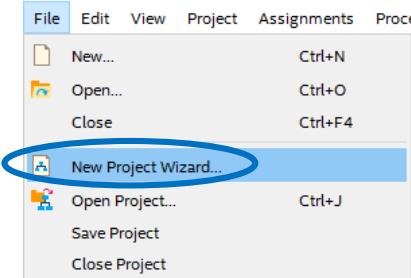
Intel® Quartus® Prime Software Projects

Intel® Quartus® Prime Design Software Projects

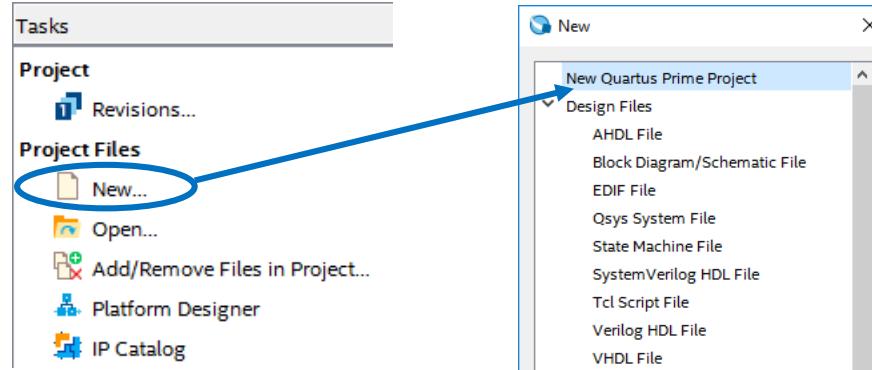
- Description
 - Collection of related design files & libraries
 - Must have a designated top-level entity
 - Target a single device
 - Store settings in Intel® Quartus® Prime software settings file (**.qsf**)
 - Compiled netlist information stored in **qdb** folder in project directory
- Create new projects with **New Project Wizard**
 - Can be created using Tcl scripts

New Project Wizard

File menu



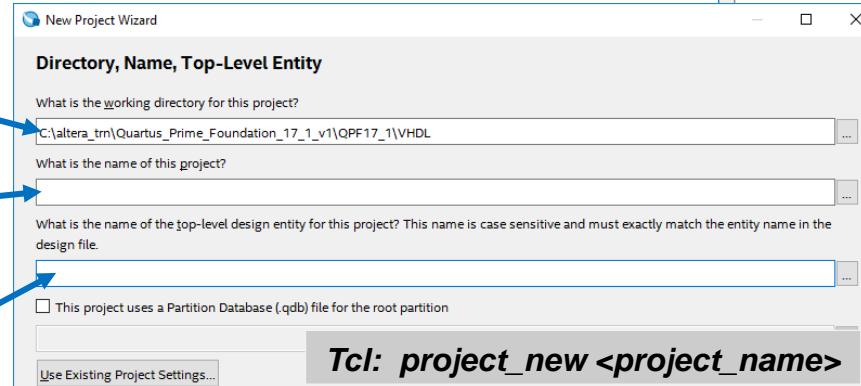
Tasks window



Select working directory

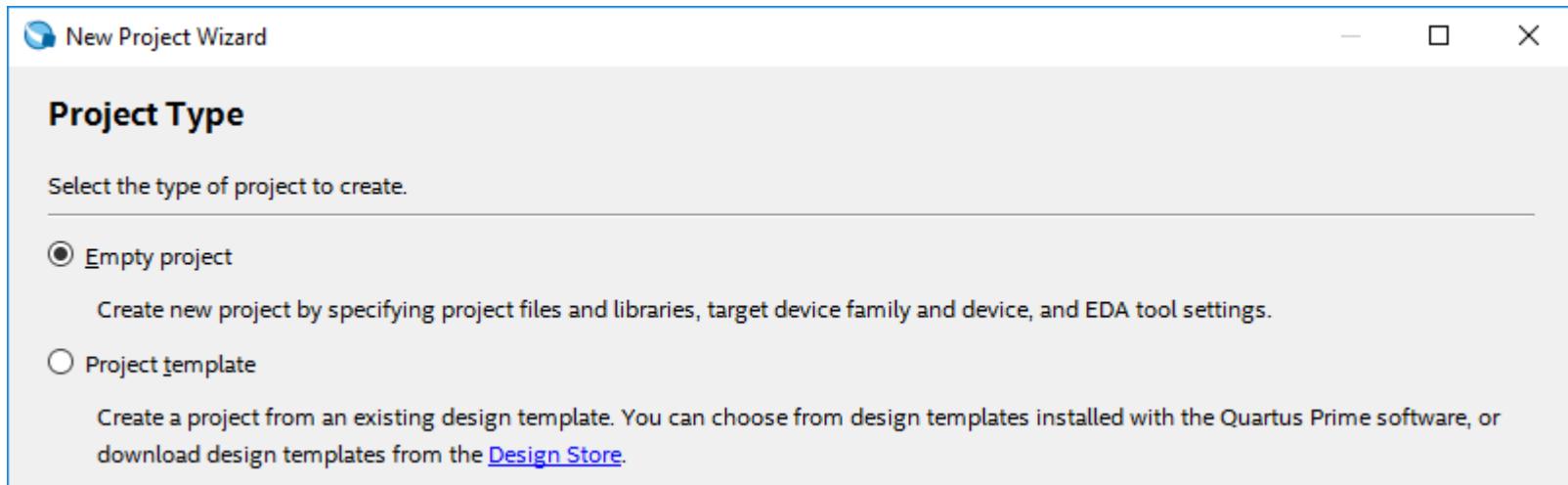
Name of project can be any name;
recommend using top-level file
name

Top-level entity does not need to be
the same name as top-level file
name



Project Type

- Create a blank project or use templates from Intel® FPGA Design Store
 - <https://cloud.altera.com/devstore/>



Intel® FPGA Design Store

Login

- Download complete example design templates for specific development kits
- Design examples include design files, device programming files, and software code as required
- Install .par files and select as template in New Project Wizard

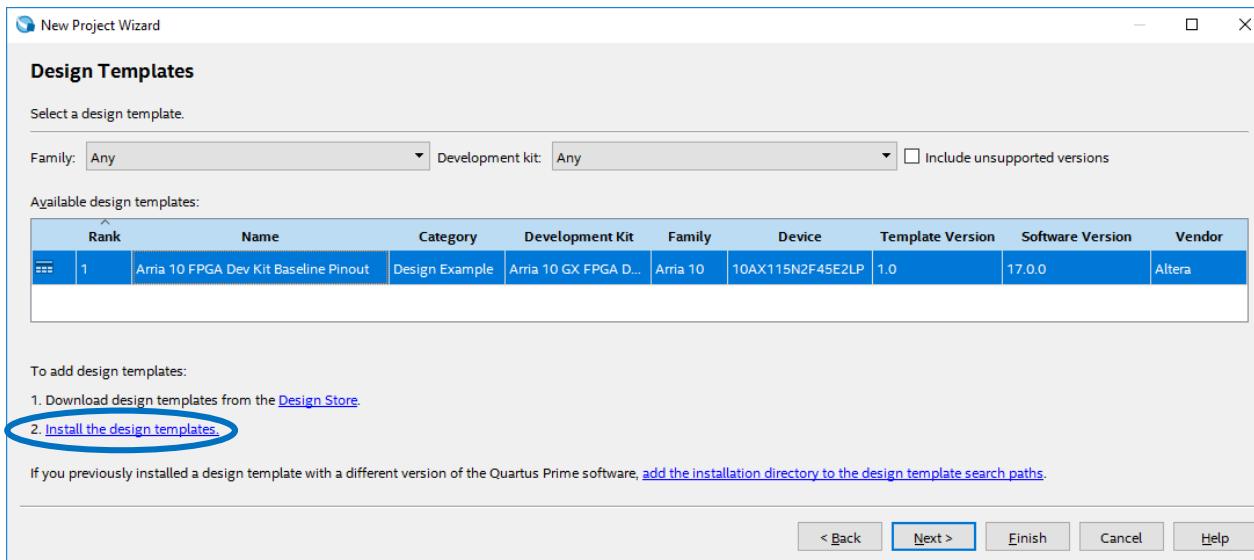
<https://cloud.altera.com/devstore/platform/>

The screenshot shows the 'Design Examples' section of the Intel FPGA Design Store. At the top, there are search filters for Family (Any), Category (Any), Quartus II Version (16.0), Development Kit (Any), and IP Core (Any). Below the filters is a search bar with 'Search' and 'Search in all pages' options. A yellow banner at the top provides links for finding more design examples and contributing content. The main area displays a table of design examples:

Name	Category	Development Kit	Family	Quartus II Version	Vendor	Downloads
JPEG Decoder Design Example (OpenCL)	Design Example \ Outside Design Store	Non kit specific Stratix V Design Examples	Stratix V	16.0.0	Altera	0
100Gbps Ethernet PHY only Testbench	Design Example \ Outside Design Store	Non kit specific Stratix V Design Examples	Stratix V	16.0.2	Altera	0
Accelerated FIR with Built-In Direct Memory Access Example	Design Example	Cyclone V E FPGA Development Kit	Cyclone V	16.0.0	Altera	81
Adapting Digilent PmodCLP LCD to DE10 Lite Development Kit Arduino Shield Header	Design Example	MAX 10 DE10 - Lite	MAX 10	16.0.0	Altera	49

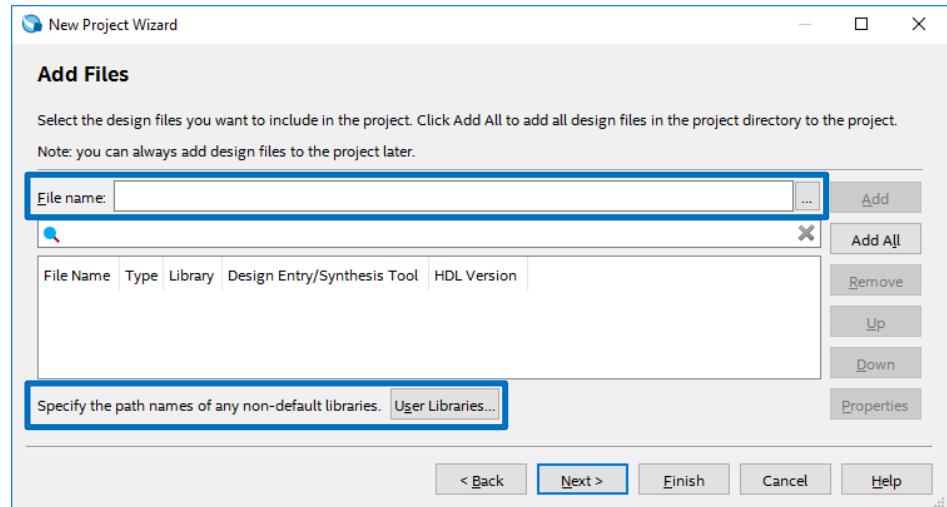
Install and Select Design Templates

- **devplatforms** folder created at **.par** installation location
- Templates available for all future project creation



Add Files

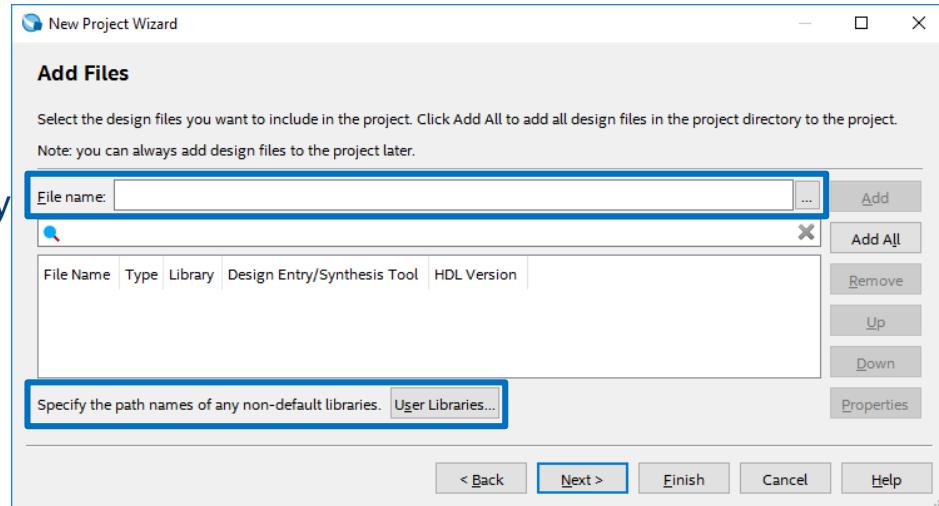
- Add design files
 - Graphic
 - VHDL
 - Verilog
 - SystemVerilog
 - EDIF
 - VQM
 - Intel® Quartus® Prime software IP
 - Platform Designer



```
Tcl: set_global_assignment -name VHDL_FILE <filename.vhd>
Tcl: set_global_assignment -name USER_LIBRARIES <library_path_name>
```

Add Files (2)

- Add library paths
 - User libraries
 - Intel® FPGA Intellectual Property (IP) library
 - Pre-compiled VHDL packages



```
Tcl: set_global_assignment -name VHDL_FILE <filename.vhd>
Tcl: set_global_assignment -name USER_LIBRARIES <library_path_name>
```

Device Selection

Choose device family
& family category
(transceiver options,
SoC options, etc.)

New Project Wizard

Family, Device & Board Settings

Device Board

Select the family and device you want to target for compilation. You can install additional device support with the Install Devices command on the Tools menu. To determine the version of the Quartus Prime software in which your target device is supported, refer to the [Device Support List](#) webpage.

Device family

Family: Arria 10 (GX/SX/GT)

Device: Arria 10 GX

Show in 'Available devices' list

Package: FBGA

Pin count: 1932

Core speed grade: 1

Transceiver speed grade: 2

Name filter:

Show advanced devices

Target device

Specific device selected in 'Available devices' list

Other: n/a

Available devices:

Name	Core Voltage	AI M	Total I/Os	GPIOs	HSSI Channels	PCIe Hard IP Blocks	Memory Bits	M20K	
10AX115S2F45I1SG	0.9V or 0.95V	427200	960	624	72	4	55562240	2713	1
10AX115S2F45I1SG...	0.95V	427200	960	624	72	4	55562240	2713	1
10AX115U2F45E1SG	0.9V or 0.95V	427200	928	480	96	4	55562240	2713	1
10AX115U2F45I1SG	0.9V or 0.95V	427200	928	480	96	4	55562240	2713	1

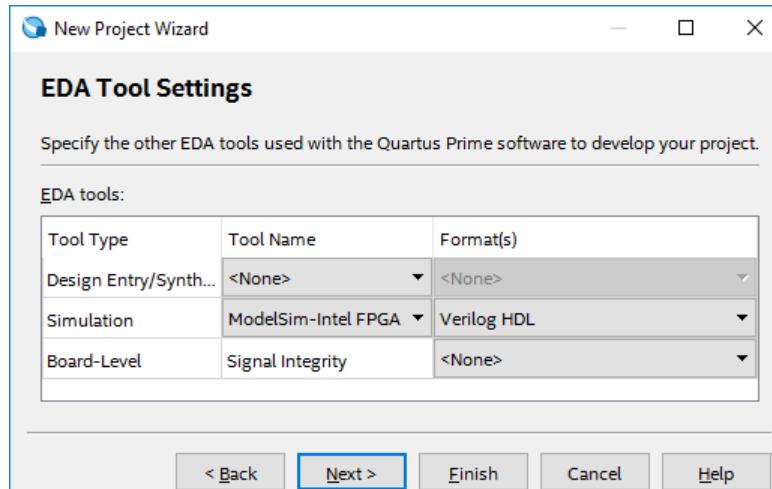
Tcl: set_global_assignment -name FAMILY "device family name"

Tcl: set_global_assignment -name DEVICE <part_number>

Filter device list

EDA Tool Settings

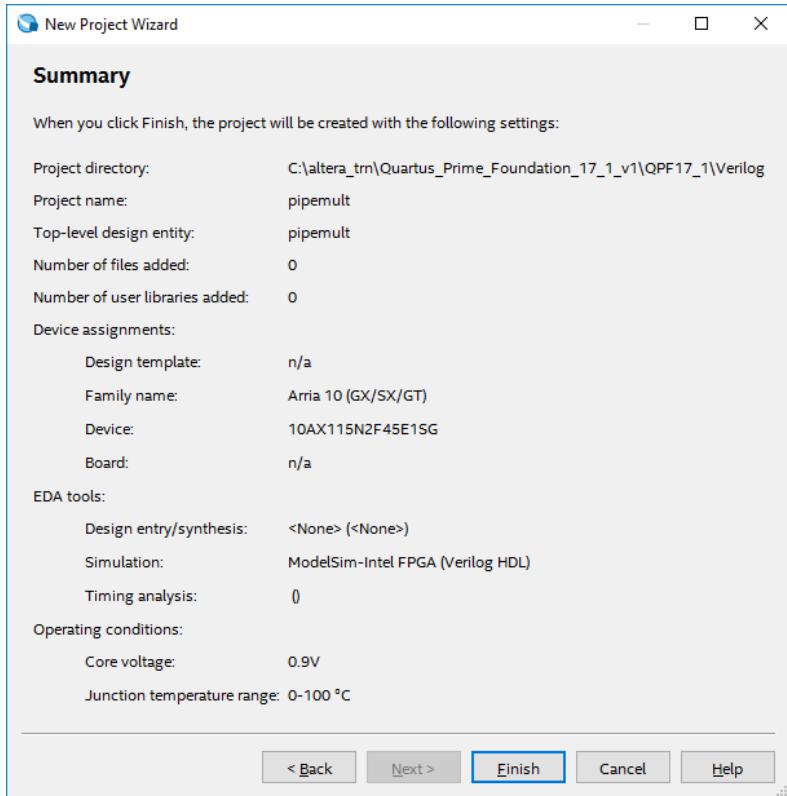
- Choose EDA tools and file formats
- Settings can be changed or added later



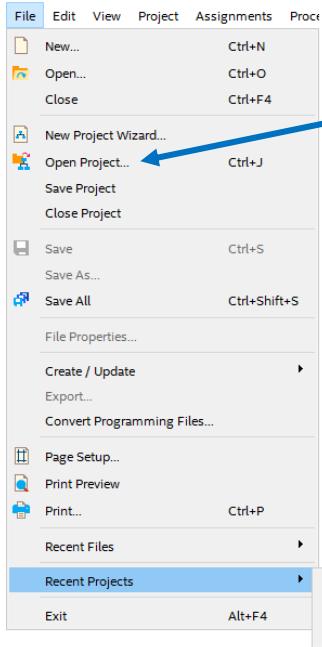
See handbook for Tcl command format

Done!

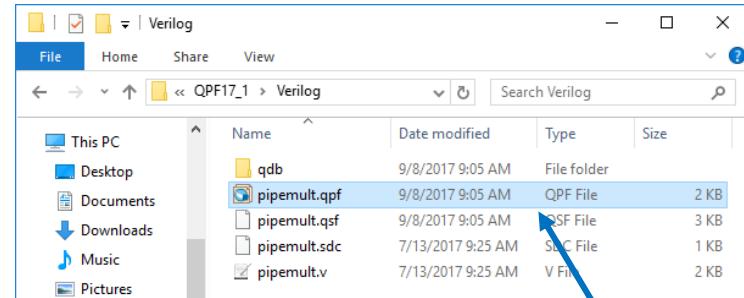
- Review results & click **Finish** when done



Opening an Existing Project



From File menu



Double-click .qpf file

Select from recent projects

Home screen

Recent Projects

- pipemult.qpf (C:/altera_trn/Quartus_Prime_Foundation_17_1_v1/QPF17_1/Verilog/pipemult.qpf)
- pipemult.qpf (C:/fpga_trn/Quartus_Prime_Software_Foundation/QPF17_1/Verilog/pipemult.qpf)
- ddr3_x72_2inst.qpf (C:/fpga_trn/Quartus_Prime_Software_Foundation/QPF17_1/InterfacePlanner/ddr3_x72_2inst.qpf)
- ddr3_x72_2inst.qpf (C:/altera_trn/Quartus_Prime_Foundation_17_1_v1/QPF17_1/test/ddr3_x72_2inst.qpf)

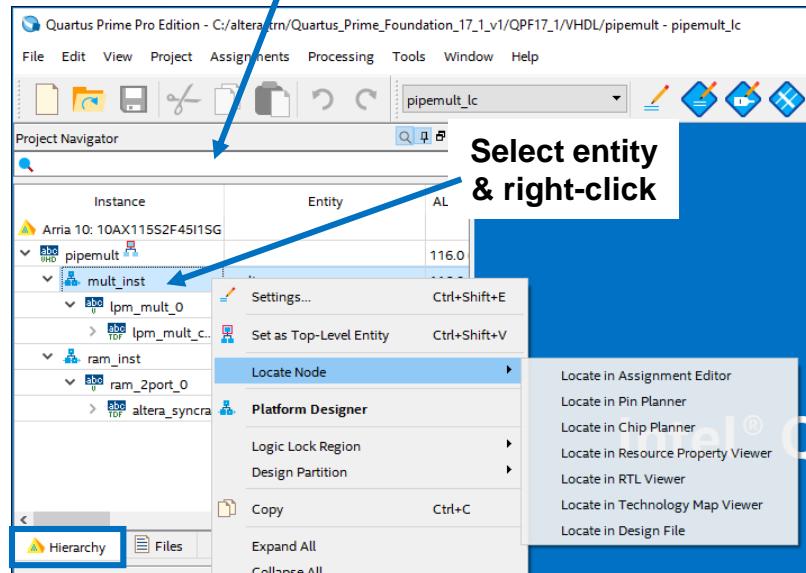
Tcl: `project_open <project_name>`

Project Navigator: Hierarchy

Displays project hierarchy after project is analyzed

- Uses
 - Set top-level entity
 - Make entity-level assignments
 - View resource usage
 - Locate in design file or viewers/floorplans (cross-probing)

Filter project hierarchy

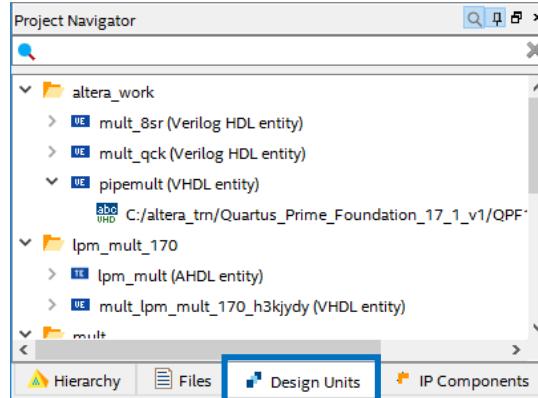
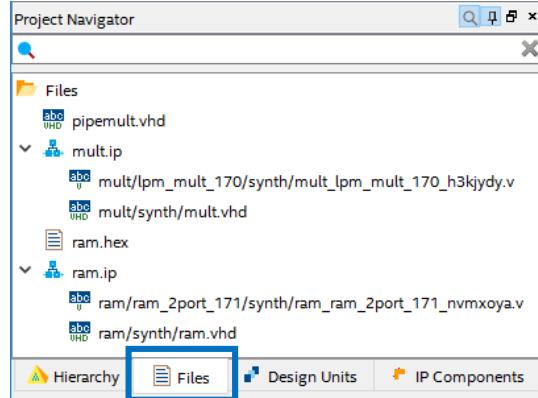


Full compilation or
Processing menu → Start → Start Analysis & Elaboration

Project Navigator: Files & Design Units (1)

Files

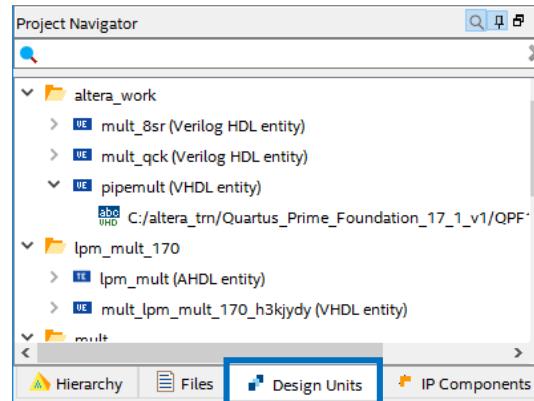
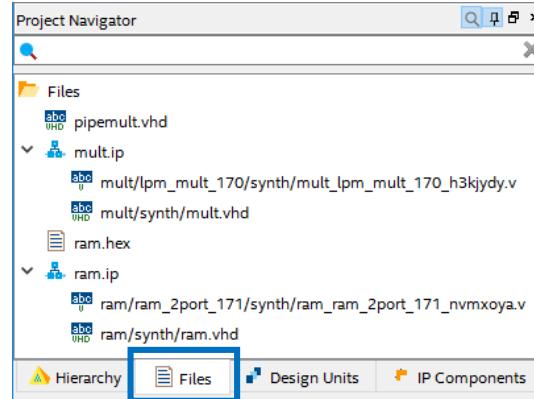
- Shows files explicitly added to project
 - Open files
 - Remove files from project
 - Set new top-level entity
 - Specify VHDL library
 - Select file-specific synthesis tool
- Can also use **Project** menu → **Add/Remove Files in Project...**



Project Navigator: Files & Design Units (2)

Design Units

- Displays design unit & type
 - VHDL entity and architecture
 - Verilog module
 - AHDL (Altera HDL) subdesign
 - Block diagram filename
- Expanded unit displays file which instantiates design unit



Intel® Quartus® Prime Design Software Project Files & Folders

- Intel® Quartus® Prime Design Software Project File (**.qpf**)
- Intel Quartus Prime Software Defaults File (**.qdf**)
- Intel Quartus Prime Software Settings File (**.qsf**)
- Synopsys* Design Constraints (**.sdc**)
 - Contains timing constraints
 - Learn about SDC and using the Timing Analyzer
 - <http://www.altera.com/education/training/courses/ODSW1115>
- **qdb** folder
 - Contains compiled design information
- **output_files** folder (customize location/name in project settings)
 - Generated compilation report files
 - Programming files generated by the Intel Quartus Prime Software Assembler

Project & Defaults Files

- Intel® Quartus® Prime Design Software Project File (**.qpf**)
 - Intel Quartus Prime software version
 - Time stamp
 - Active revision(s)
 - *Discussed in a moment*
- Intel Quartus Prime Software Defaults Files (**.qdf**)
 - Stores Intel Quartus Prime software project setting & assignment defaults for new project revisions
 - *<revision_name>_assignment_defaults.qdf*
 - Found in local project or *altera_pro\<version>\quartus\bin64* directory (*assignment_defaults_pro.qdf*)

pipemult.qpf

```
QUARTUS_VERSION = "17.1"
DATE = "09:05:07 September 08, 2017"

# Revisions

PROJECT_REVISION = "pipemult"
```

Intel® Quartus® Prime Design Software Settings File (.qsf)

- Stores all settings & assignments except timing
- Uses Tcl syntax
- Can be edited manually by user
 - See the Intel® Quartus® Prime Design Software Settings File Reference Manual (https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/manual/mnl_qsf_reference.pdf) for more details on QSF assignments & syntax

Source other Tcl/.qsf files to organize assignments

User comments start with #

```
# Quartus Prime
# Version 17.0.0 Build 290 04/26/2017 SJ Pro Edition
# Date created = 07:57:46 July 14, 2017
#
# -----
# Notes:
#
# 1) The default values for assignments are stored in the file:
#     pipemult_lc_assignment_defaults.qdf
#     If this file doesn't exist, see file:
#     assignment_defaults.qdf
#
# 2) Intel recommends that you do not modify this file. This
#     file is updated automatically by the Quartus Prime software
#     and any changes you make may be lost or overwritten.
#
# -----
source "location_assignments.tcl"
set_global_assignment -name FAMILY "Arria 10"
set_global_assignment -name DEVICE 10AX115S2F45I1SG
set_global_assignment -name TOP_LEVEL_ENTITY pipemult
set_global_assignment -name ORIGINAL_QUARTUS_VERSION 17.0.0
set_global_assignment -name PROJECT_CREATION_TIME_DATE "18:15:20 JULY 16, 2017"
set_global_assignment -name LAST_QUARTUS_VERSION "17.1.0 Pro Edition"
set_global_assignment -name VHDL_FILE pipemult.vhd
```

Constraint Files & Assignment Priority

- **.qsf**
 - Highest priority
 - Assignments always used from here first
- Revision-specific **.qdf** file located in project directory
 - *<revision_name>_assignment_defaults.qdf*
 - Created automatically in the project directory when a revision is opened in another version of the Intel® Quartus® Prime software
- **.qdf** located in project directory
 - *assignment_defaults.qdf*
 - Created automatically in project directory when project archived & restored
- **.qdf** located in Intel Quartus Prime Design Software **bin64** directory
 - Lowest priority
 - Assignments only used if not found in higher priority files

Project & IP Management

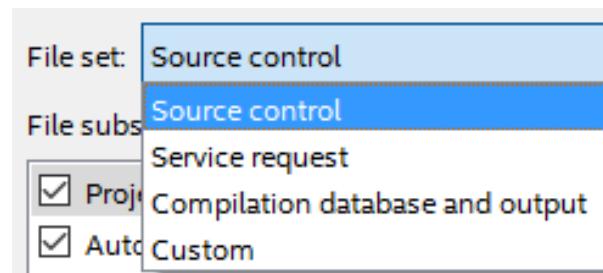
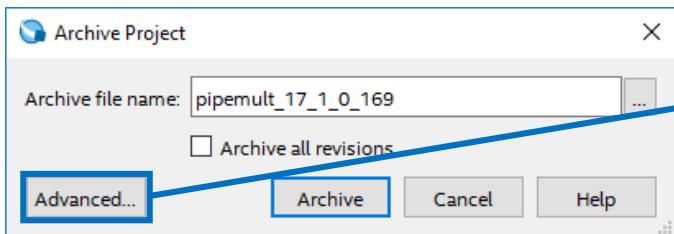
- Project archive & restore
- Project copy
- Revisions
- Project clean
- IP management tools

Project Archive (1)

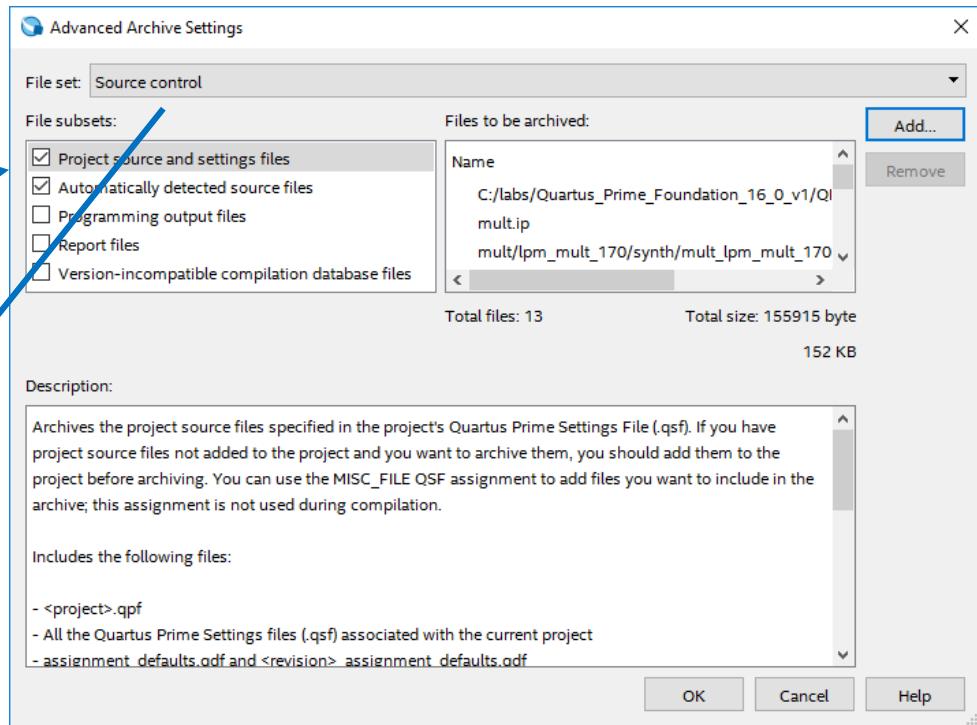
- Creates 2 files
 - Compressed Intel® Quartus® Prime Design Software Archive File (**.qar**)
 - Includes design files, **.qpf** file, & **.qsf** file(s)
 - Option to include databases
 - Creates local **.qdf** file for archive
 - Archive activity log (**.qarlog**)
- Example uses
 - File storage (revision control)
 - Project handoff: useful for sending to Intel® FPGA support

```
Tcl: project_archive <project_name>
```

Project Archive (2)



Project menu or Tasks window

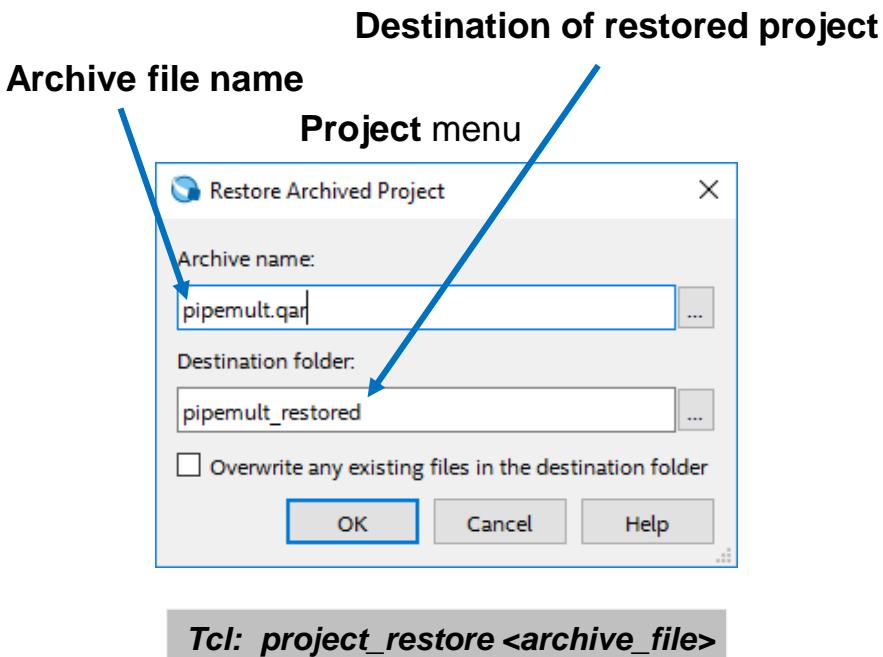


External Revision Control

- Track and protect project file revisions, especially in a team-based design
- Files to include (in a **.qar** or independently)
 - Design files (**.v**, **.sv**, **.vhd**, **.bdf**, **.edf**, **.vqm**)
 - Timing constraints (**.sdc**)
 - Project settings and assignments (**.qdf**, **.qpf**, **.qsf**)
 - IP files (**.v**, **.sv**, **.vhd**, **.qip**, **.sip**, **.qsys**)
 - Platform Designer-generated files (**.qsys**, **.qip**, **.sip**)
 - Files for 3rd-party tools (**.vo**, **.vho**)
- If not using **.qar**, use Advanced Archive Settings as a guide for files to store

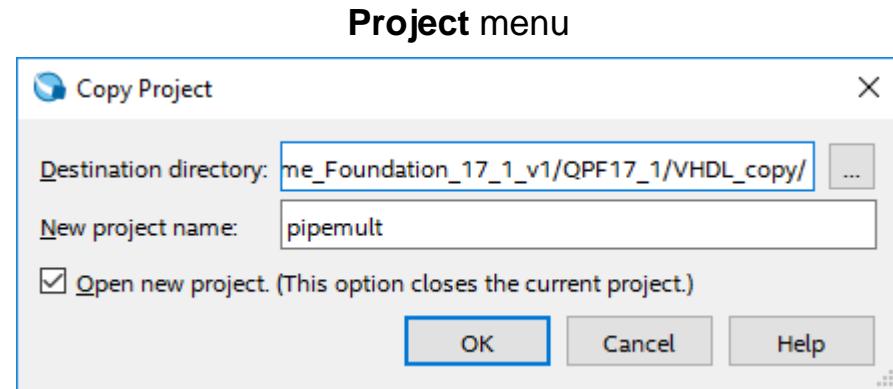
Restore Archived Project

- Decompresses .qar into specified directory
- Paths/directory structures to referenced files/libraries *outside* project directory must also be restored
 - Recreated in restore location based on nearest common *parent directory*
 - Example of referenced file paths in restored project destination:
 - <destination folder>/drive/C/<entire path to all files in project directory>
 - <destination folder>/drive/H/<path to file(s) referenced on original H drive>



Project Copy

- Copy & save *exact* duplicate of project in new directory
 - Project file (**.qpf**)
 - Design files
 - Settings files
- Example use: duplicating work before editing design files
- User libraries are **not** copied; check paths
- New local **.qdf** not created; only copies **.qdf** if it exists



Revisions

- Explore new sets of constraints
- Compile options without losing previous work
- Revision-specific project files generated at next compilation and stored in qdb directory

Creating a Revision

**Project menu → Revisions
or Tasks window**

Current revision

Create new revision

Select revision and click to set current revision

Base revision on any previous revision

Tcl: *create_revision <revision_name>*

Revisions:

Revision Name	Revision Type	Top-level Entity	Family	Device	Timing Model	Description
pipemult_lc	pipemult	Arria 10	10AX115S2F45I1SG			Created on:Sunday, July 16, 2017 Based on: pipemult
pipemult	pipemult	Arria 10	10AX115S2F45I1SG			
<<new revision>>						

Create Revision

Specify a name and description for the new revision. You can base the revision on an existing revision, and specify the revision as the current revision.

Revision name:

Based on revision:

Description:

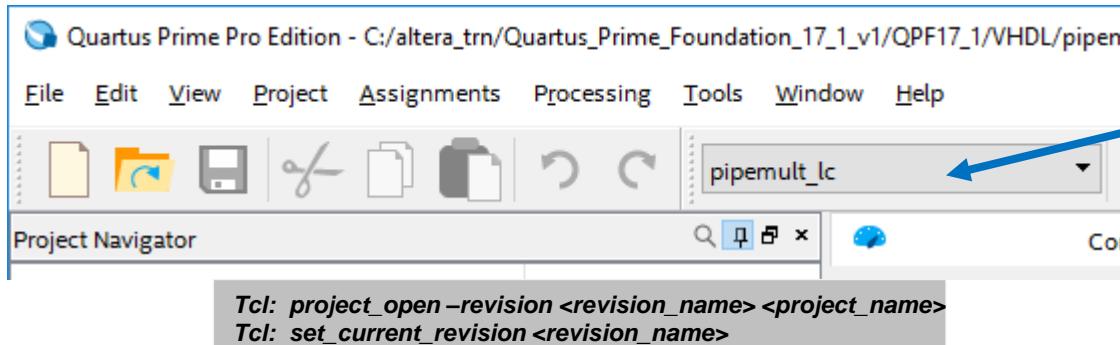
Created on: Friday, September 08, 2017
Based on: pipemult_lc

Type revision description (optional)

Set as current revision

Project Revision Support

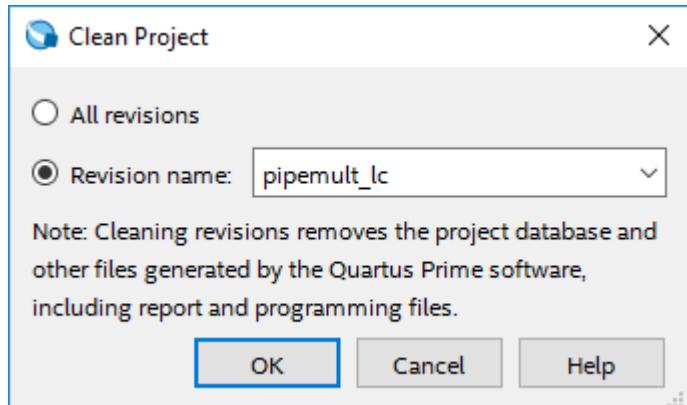
- **.qsf** created for each revision
 - *<revision_name>.qsf*
- Active revision names stored in **.qpf**
- Text file created for each revision (from description field)
 - *<revision_name>_description.txt*



Quickly switch
between revisions

Project Clean

- Cleans databases and output files generated for a given revision (or revisions)
 - Removes revision files in project subfolders
 - Removes reports, programming files, and other compiler-generated output
- Select **Clean Project** from the **Project** menu
 - Select from available revisions
 - Clean multiple revisions using wildcards



Tcl: project_clean –revision <revision_name> <project_name>

IP Management Tools

- Keep track of project IP cores
- Flag out-of-date IP cores
- Upgrade IP cores to latest version

Project Navigator: IP Components

- View all IP in project
 - Component name/file, Intel® Quartus® Prime Design Software version, vendor
- View if new version of IP available
 - Required update
 - Optional update
 - IP is up to date
- Right-click to upgrade or edit individual IP

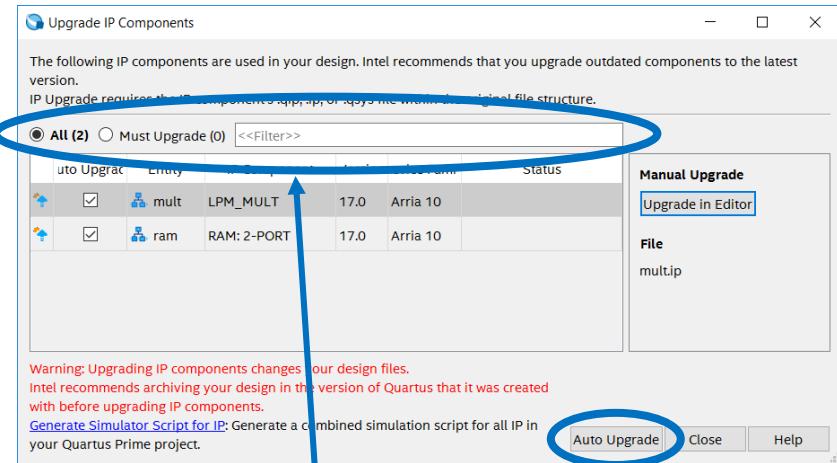
The screenshot shows the Project Navigator window with a yellow banner at the top stating "IP upgrade recommended." and a "Launch IP Upgrade Tool..." button. Below the banner is a table with two rows of IP components. The first row contains "mult" and "LPM_MULT" with version "17.0" and supported device "Arria 10". The second row contains "ram" and "RAM: 2-PORT" with version "17.0" and supported device "Arria 10". A blue arrow points from the text "Filter IP list" to the "Launch IP Upgrade Tool..." button. Two blue circles highlight the "mult" row and the "Version" column header. A blue arrow points from the text "Icon indicates new version is available; double-click to update manually through parameter editor" to the "Version" column of the "mult" row.

Entity	IP Component	Version	Supported Device
mult	LPM_MULT	17.0	Arria 10
ram	RAM: 2-PORT	17.0	Arria 10

Icon indicates new version is available; double-click to update manually through parameter editor

Upgrading IP Components

- Software detects out-of-date IP when opening existing project
 - Manually open from **Project** menu, Project Navigator, or IP right-click
 - Displays required and optional upgrades
- Select IP block(s) and click **Perform Automatic Upgrade** if supported
 - Platform Designer systems and components must be regenerated in Platform Designer



**Filter by name or upgrade status
(Required or optional upgrade)**

Exercise 1

Projects Summary

- Projects necessary for design processing
- Use New Project Wizard to create new projects
- Use Project Navigator to study file & entity relationships within project
- Project archive, copy, revisions, and clean provide easy-to-use project management

Project Support Resources

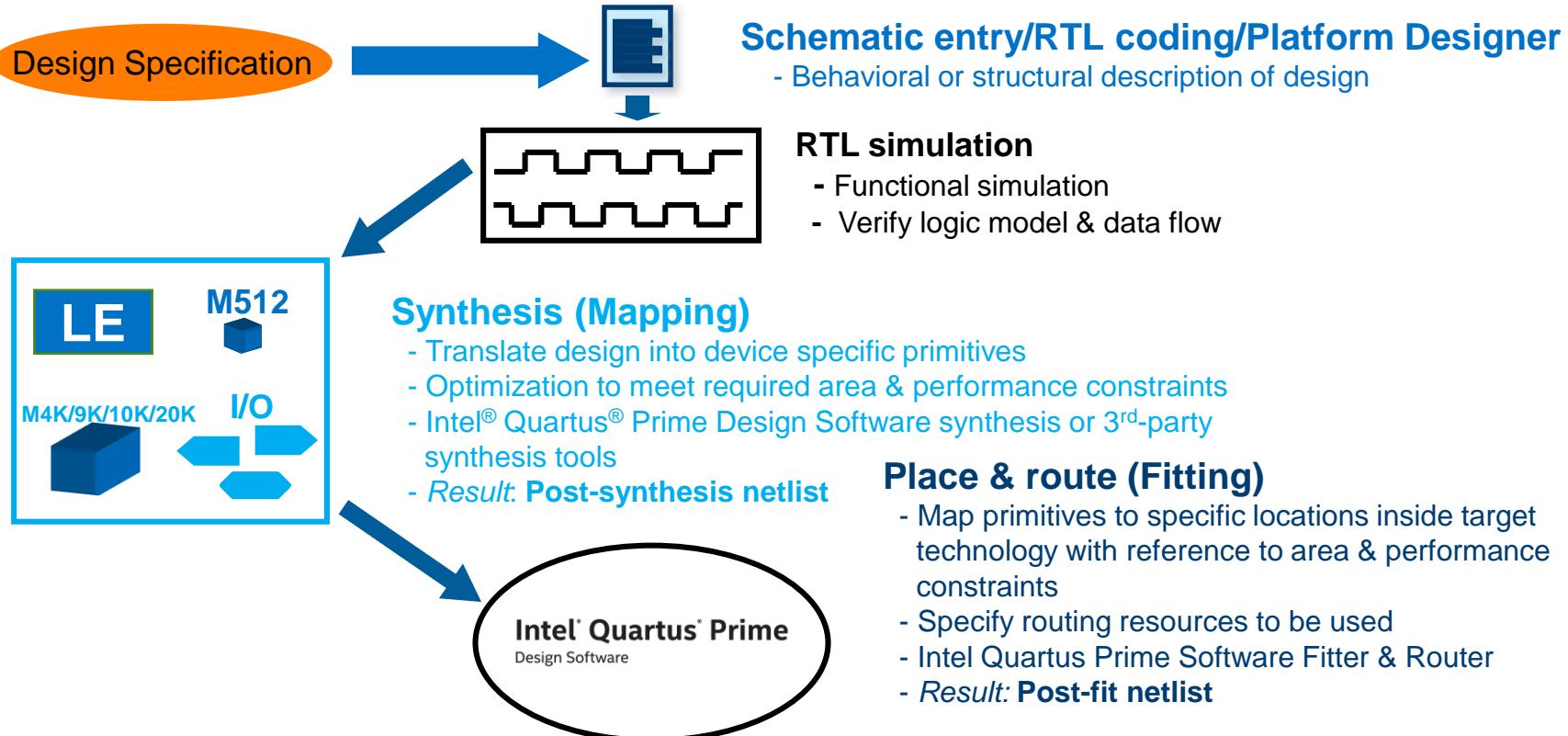
- Managing Intel® Quartus® Prime Design Software Projects chapter in Volume 1 of the Intel Quartus Prime Software Pro Handbook
- Intel Quartus Prime Software Handbook always available online in pdf format at: <https://www.altera.com/products/design-software/fpga-design/quartus-prime/support.html>



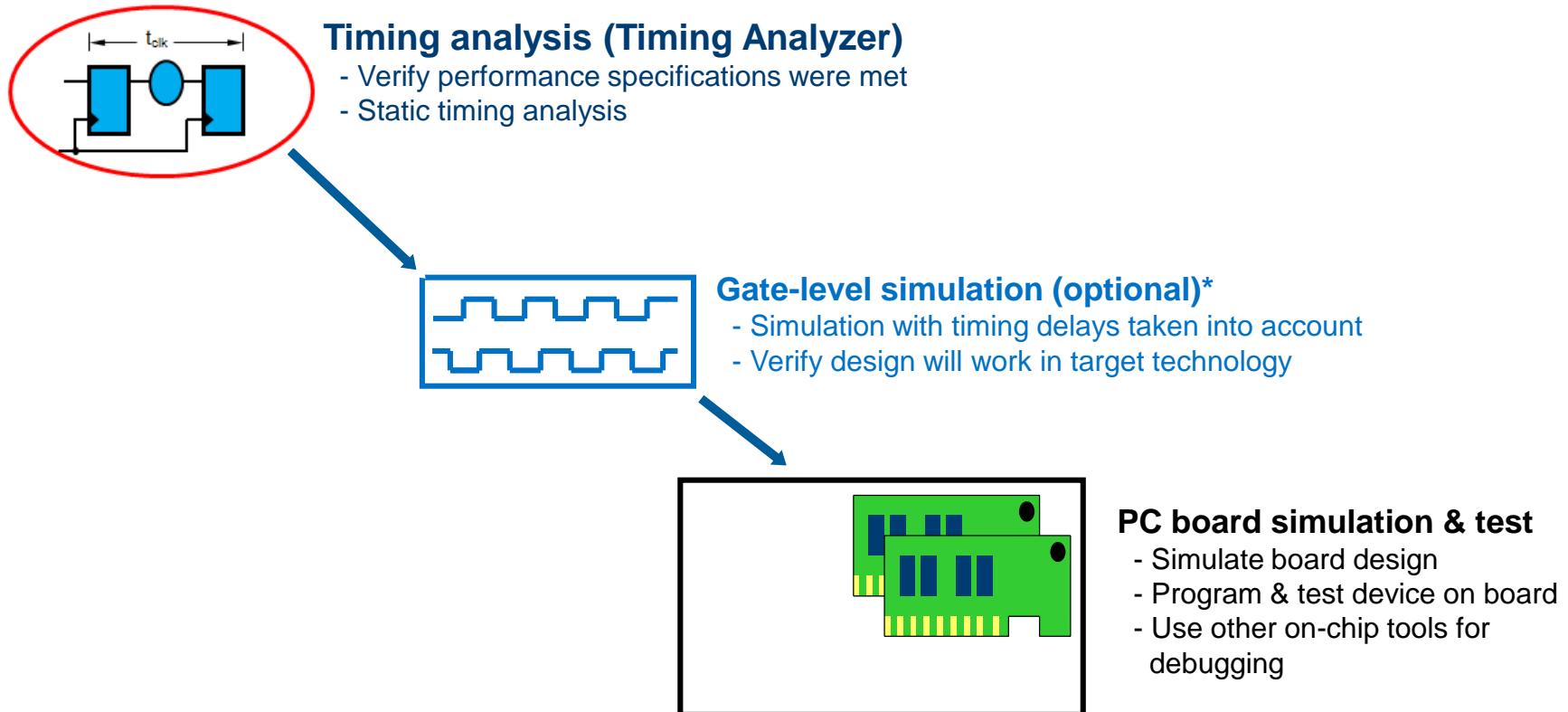
The Intel® Quartus® Prime Design Software: Foundation

Design Methodology

Typical PLD Design Flow



Typical PLD Design Flow



* Not supported in 20-nm and newer devices



The Intel® Quartus® Prime Design Software: Foundation

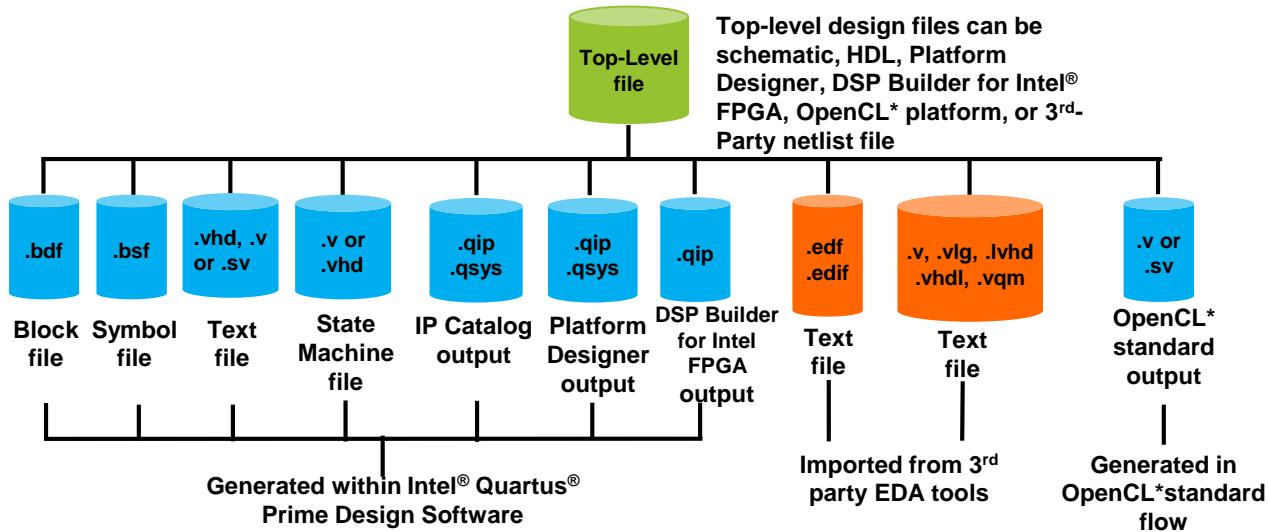
Design Entry

Design Entry Methods

- Intel® Quartus® Prime Design Software design entry
 - Text editor
 - VHDL
 - Verilog or SystemVerilog
 - Schematic editor
 - Block Diagram File
 - System editor
 - Platform Designer
 - State machine editor
 - HDL from state machine file
 - Memory editor
 - HEX
 - MIF
- 3rd-party EDA tools
 - EDIF 2 0 0
 - Verilog Intel Quartus software Mapping (.vqm)

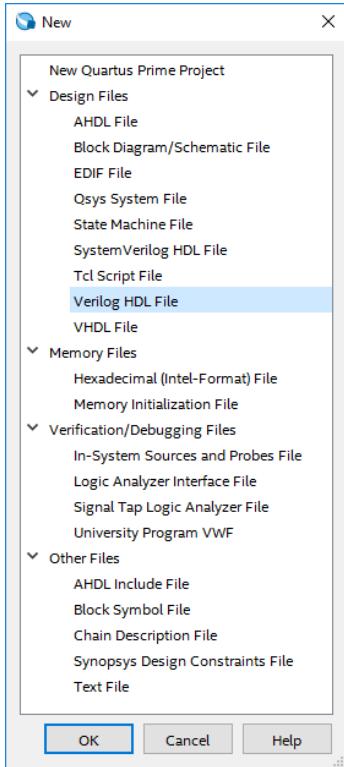
Design Entry File Types Supported

- Mixing & matching design files allowed
 - Not always the case with some 3rd-party tools
 - ModelSim*-Intel® FPGA [Starter] Edition does support mixed language simulation

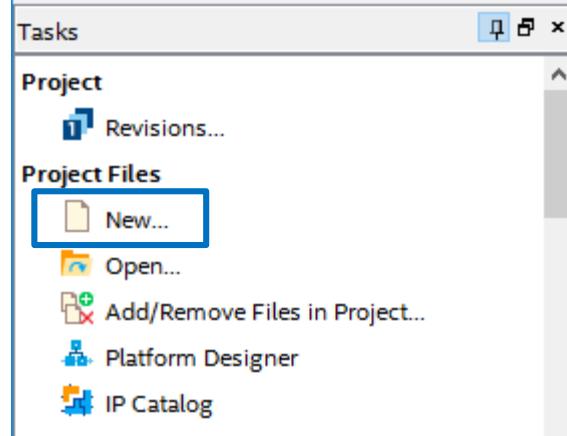


Creating New Design Files (& Others)

File menu → New or  in Toolbar



Tasks window



Text Design Entry

- Intel® Quartus® Prime Design Software
Text Editor features
 - Block commenting
 - Line numbering in HDL text files
 - Bookmarks
 - Syntax coloring
 - Find/replace text
 - Find and highlight matching delimiters
 - Function collapse/expand
 - Create & edit .sdc files for Timing Analyzer
 - Preview/editing of full design and construct HDL templates
- Enter text description
 - VHDL (.vhd, .vhdl)
 - Verilog (.v, .vlg, .Verilog, .vh)
 - SystemVerilog (.sv)

Note: select your default text editor from
Tools menu → **Options** → **Preferred Text Editor**

Verilog & VHDL

- **VHDL** - VHSIC hardware description language
 - IEEE Std 1076 (1987 & 1993) supported
 - Partial IEEE Std 1076-2008 support
 - IEEE Std 1076.3 (1997) synthesis packages supported
- **Verilog**
 - IEEE Std 1364 (1995 & 2001) & 1800 (SystemVerilog) supported
- Use Intel® Quartus® Prime Design Software
- View supported commands in built-in help
- Learn more about HDL in Intel® FPGA HDL customer training classes

Text Editor Features

Find/highlight matching
delimiters

A screenshot of the Quartus Prime Text Editor interface. The main window displays VHDL code for a component named 'pipemult'. A cursor is positioned at the start of the identifier 'dataa'. A tooltip box labeled 'Auto text-complete' appears, showing the completed identifier 'dataa' with a small preview of its definition. The editor's toolbar and menu bar are visible at the top.

Bookmarks
(on/off/jump to)

Insert Template



(Edit menu)

A screenshot of the 'Insert Template' dialog box. The left pane shows a list of language templates, including AHDL, Quartus Prime TCL, TimeQuest, SystemVerilog, TCL, Verilog HDL, and VHDL. Under VHDL, there are sections for 'Full Designs' and 'RAMs and ROMs'. The 'RAMs and ROMs' section is expanded, showing various RAM-related templates. The right pane displays a preview of the selected 'Quartus Prime VHDL Template' for a 'Single port RAM with single read/write address'. The preview shows the library declaration, entity definition with generic and port clauses, and an architecture clause. At the bottom of the dialog are 'Save', 'Insert', and 'Close' buttons.

Collapse/expand
functions

Preview window: edit before
inserting & save as user template

Additional Text Editor Features

- Auto-complete
- Smart highlighting
- Syntax color differentiation
- Error message indicator & tooltips

The screenshot shows a Verilog code editor interface. A portion of the code is visible:

```
BEGIN
    rx_locked      <= sub_wire0;
    rx_out        <= sub_wire1(7 DOWNTO 0);
    rx_outclock   <= sub_wire2;

    rx|
    rx_align_data_reg : ALTLVDS_RX
    rx_in          : ALTLVDS_RX
    rx_inclock
    rx_locked
    rx_out         : ALTLVDS_RX
    rx_outclock
    common_rx_tx_pll => "OFF",
    data_align_rollover => 4,
    data_rate => "UNUSED",
    deserialization_factor => 4,
    dpa_initial_phase_value => 0,
```

The code uses syntax highlighting where keywords like `BEGIN`, `input`, and `reg` are in blue. Variable names like `rx`, `rx_locked`, and `rx_out` are in black. Type annotations like `: ALTLVDS_RX` are in green. The code editor also features auto-complete, as evidenced by a dropdown menu appearing over the `rx` identifier.

The screenshot shows a Verilog code editor interface with several error messages indicated by red circles with an 'X' and a line number. The code is:

```
1 //18 bit complex input, 36 bit real and imaginary outputs
2 // (a0+j*b0)*(a1+j*b1) = (a0*a1-b0*b1)+j*(a0*a1+b0*b1)
3 module half_dsp_block (clock, areset, clock_ena, a0, b0, a1, b1, rout, iout)
4     parameter WIDTH = 18;
5     error(10161); Verilog HDL error at half_dsp_block.v(3): object "clock" is not declared
6     input areset;
7     input clock_ena;
8     input signed [WIDTH-1:0] a0;
9     input signed [WIDTH-1:0] b0;
```

The errors are:

- Line 3: `error(10161); Verilog HDL error at half_dsp_block.v(3): object "clock" is not declared`

Schematic Design Entry

- Full-featured schematic design capability
- Schematic Editor uses
 - Create simple test designs to understand the functionality of an Intel® FPGA IP: PLL, LVDS I/O, memory, etc...
 - Create top-level schematic for easy viewing & connection
 - Convert between schematic .bdf, block symbol .bsf, and HDL files
- Note: Schematic entry online training available: Using the Intel® Quartus® Prime Software: Schematic Design
(<http://www.altera.com/education/training/courses/ODSW1105>)

Intel® FPGA IP Cores

- Pre-made design blocks
- Benefits
 - Configurable, parameterized settings add flexibility & portability
 - “Drop-in” support to accelerate design entry
 - Pre-optimized for Intel® FPGA architecture
- Many basic non-encrypted functions included for free in web and subscription editions
- More advanced IP available in two forms
 - IP Base Suite
 - Intel® FPGA IP

IP Base Suite

- Free & installed with the Intel® Quartus® Prime Design Software
 - License required for Lite edition
 - Otherwise operate in Intel® FPGA IP Evaluation Mode (discussed in a moment)
 - HDL simulation models installed in Intel Quartus Prime libraries
- Included in Intel FPGA IP Base Suite
 - FIR Compiler
 - Numerically Controlled Oscillator
 - Fast Fourier Transform Compiler
 - AXI BFM (bus functional models) for simulation
 - DDR2/3/QDRII/LPDDR2/RDRAMII SDRAM Controllers with UniPHY
 - See <https://www.altera.com/products/intellectual-property/design/ip-base-suite.html> for details

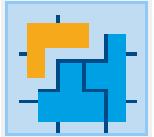
Intel® FPGA Intellectual Property (IP)

- Must purchase license (except IP Base Suite)
 - Logic for IP function is encrypted
- Two types
 - Intel® FPGA IP: developed by Intel FPGA
 - Intel® FPGA IP Partner Program
- All Intel® FPGA Intellectual Property functions & some AMPP functions support Intel® FPGA Evaluation Mode feature
 - Develop design using free version of core
 - HDL simulation models provided with IP
 - Generate time-limited, JTAG-tethered configuration/programming files
 - See [AN320: Intel® FPGA IP Evaluation Mode Evaluation of Megafunctions](#)

Best-In-Class Intel® FPGA IP Examples

- Triple-Speed Ethernet MAC
- 10Gb Ethernet MAC
- CRC Compiler
- IP Compiler for PCI Express* soft core
 - Note: Hard IP for PCI Express technology does not required a license
- Video and Image Processing Suite
- See <https://www.altera.com/products/intellectual-property/overview.html> for a complete list of Intel® FPGA IP solutions

IP Catalog



- Centralized tool for the implementation, configuration, & generation of IP cores optimized for a specific device family
- Includes listing of built-in and custom IP

No project open
Generate IP for any device family

The screenshot shows the IP Catalog window with a search bar at the top. Below it is a tree view of IP components:

- Installed IP
 - Project Directory
- Library
 - Basic Functions
 - DSP
 - Interface Protocols
 - Low Power
 - Memory Interfaces and Controllers
 - Processors and Peripherals
 - University Program

Search

Tools menu → IP Catalog or
View menu → Utility Windows → IP Catalog

Project open
Generate IP for project device or any device family

The screenshot shows the IP Catalog window for a project:

- Installed IP
 - Project Directory
- Library
 - Basic Functions
 - DSP
 - Interface Protocols
 - Low Power
 - Memory Interfaces and Controllers
 - Processors and Peripherals
 - University Program

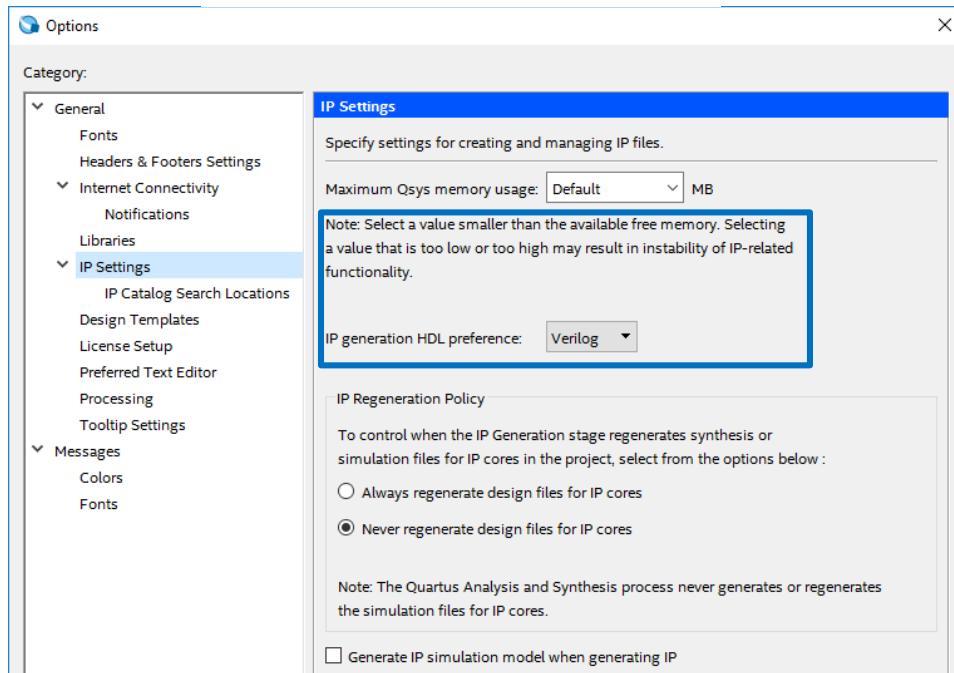
On the right, a context menu is open over the Project Directory node:

- Refresh IP catalog F5
- Show IP for all device families
- Show IP for active device family

IP Settings

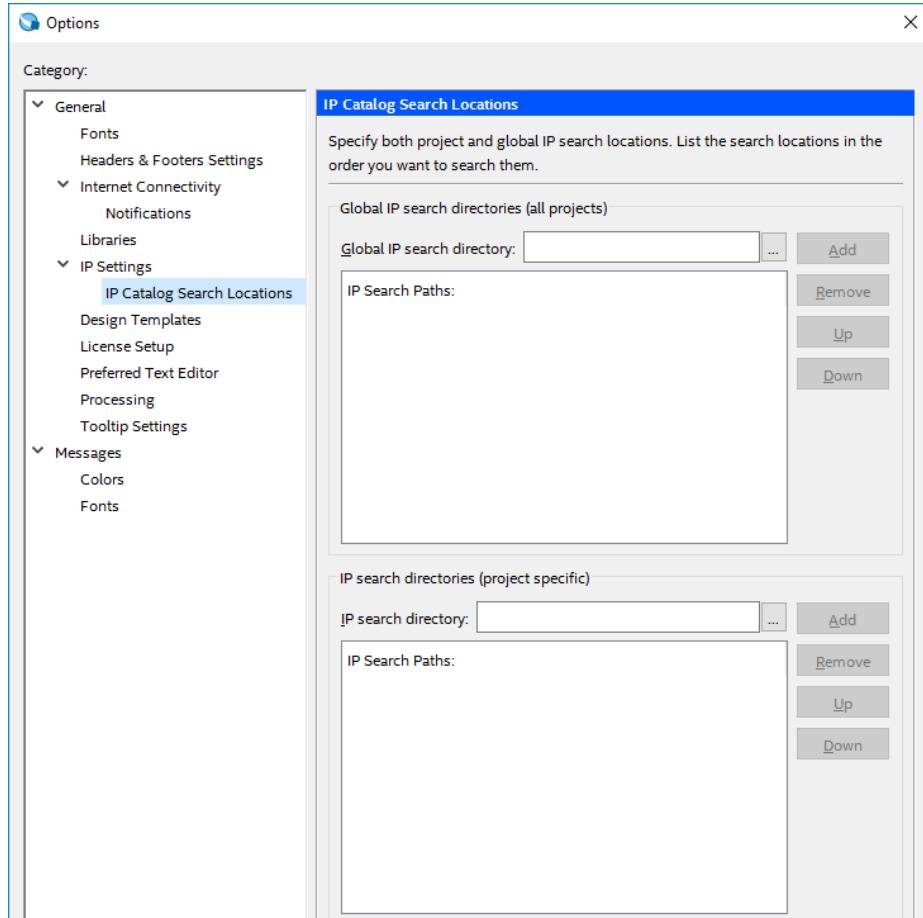
- Choose default output HDL for IP
- Automatically add generated IP to project (.qip files)

Tools menu → Options



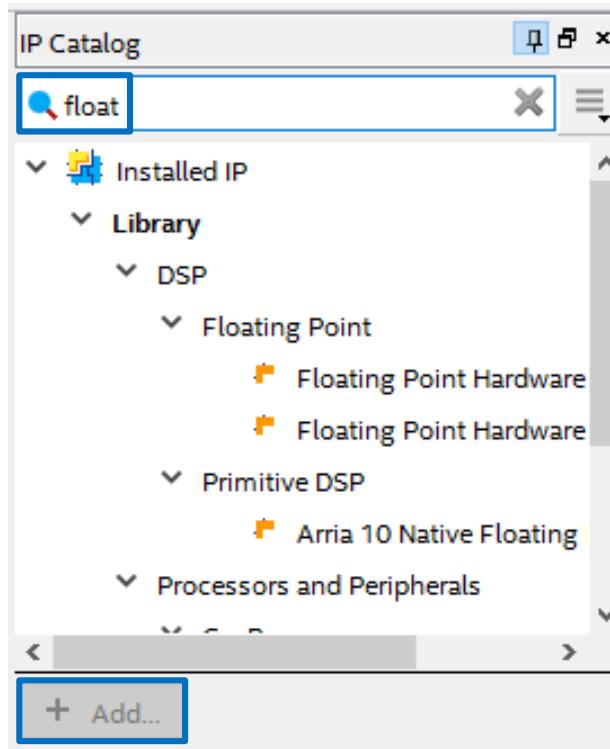
Search for IP

- Point to 3rd-party or custom IP for access in IP Catalog
- Paths shared with Platform Designer (*described later*)



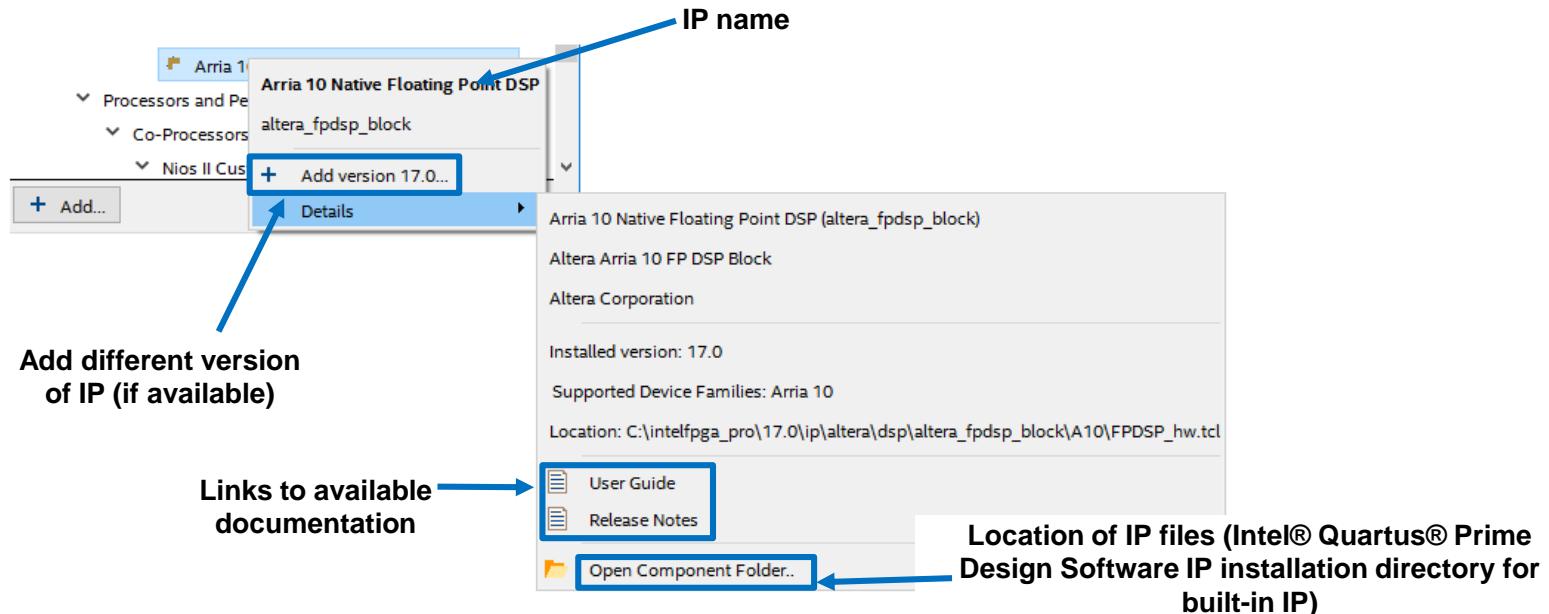
Using the IP Catalog

- Select IP to create and click Add, or double-click IP
- Name IP (for output file names)
- Specify IP parameters
- Generate IP files



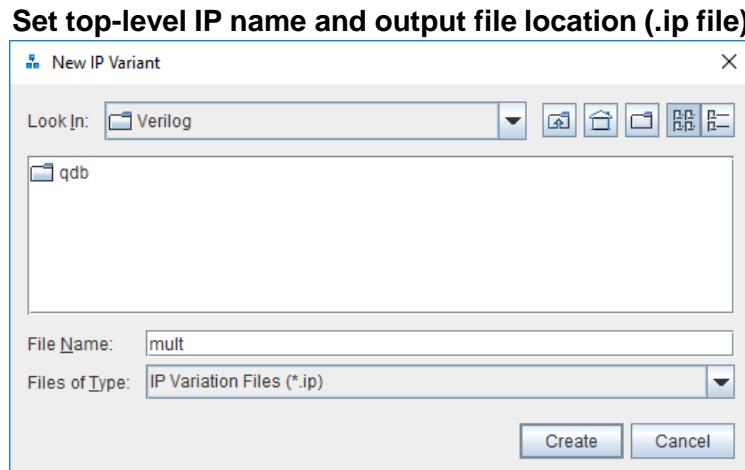
IP Options and Documentation

- Right-click IP in IP Catalog to add different versions or get help and documentation



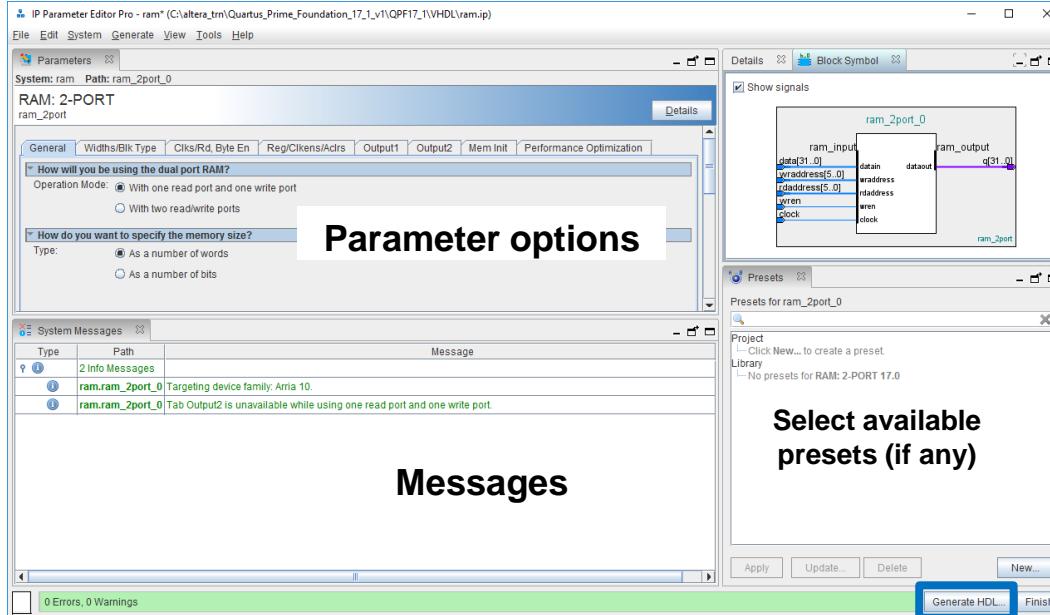
Creating an IP Variation with the IP Parameter Editor

- For all IP cores in the Intel® Quartus® Prime Design Software Pro Edition (and some in Standard Edition)
- For Intel® Arria® 10 and newer devices



IP Parameter Editor

- Generates a Platform Designer system for each parameterized IP core saved as .ip file

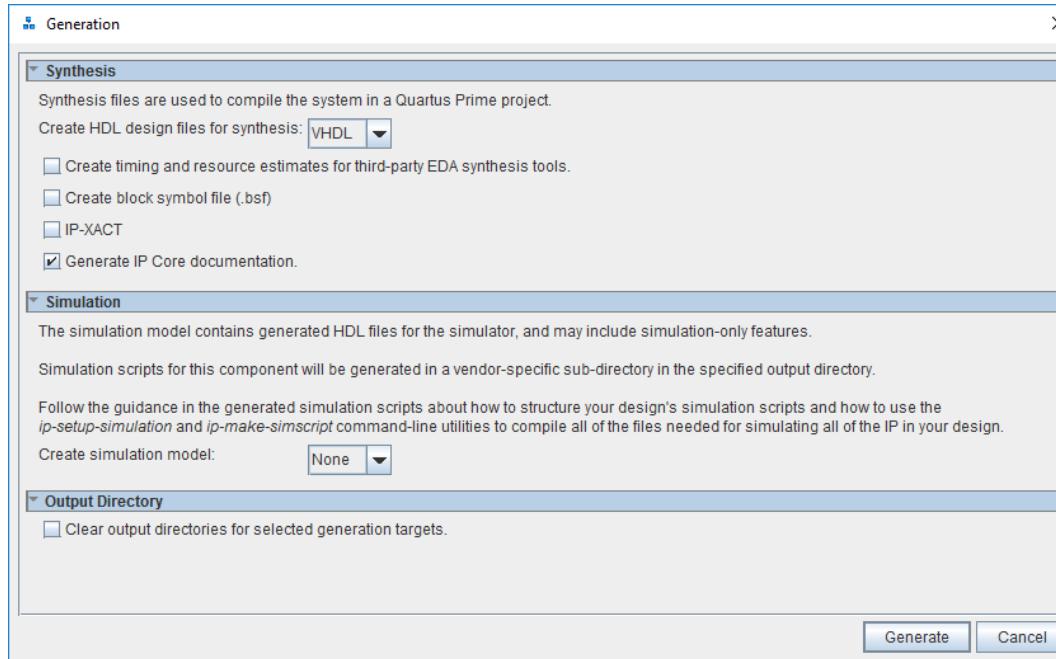


Block diagram with
signals/interfaces

Select available
presets (if any)

IP Parameter Editor: Generate HDL

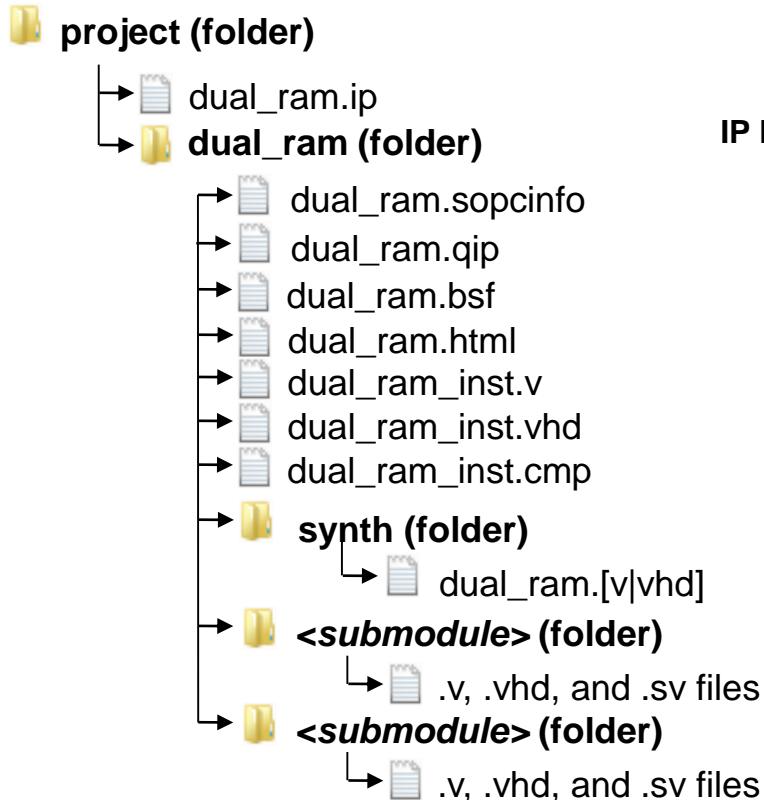
- Choose to generate files for synthesis and/or simulation



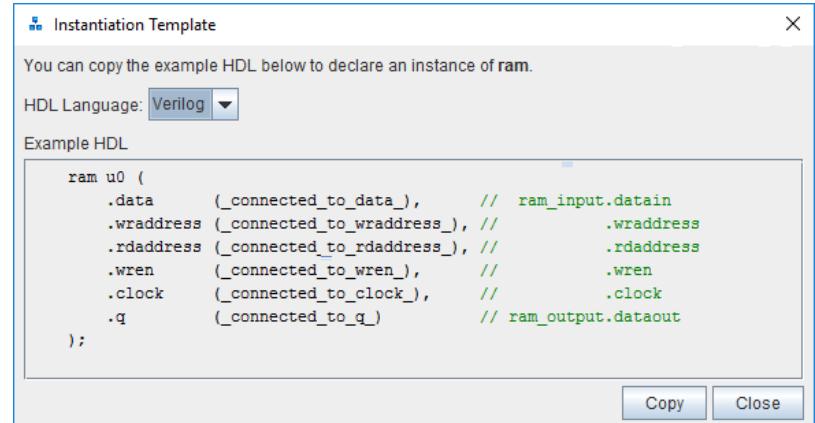
Output Files

- Top-level folder files
 - <IP_name>.ip
 - Main Platform Designer system file; can be opened using IP Parameter Editor or Platform Designer
- Located in <IP_name> folder
 - <IP_name>.sopcinfo
 - XML file describing IP used for software development tools
 - <IP_name>.qip
 - Script file that adds all files needed for synthesis to Intel® Quartus® Prime Design Software project
 - <IP_name>.bsf
 - Symbol file for Intel Quartus Prime schematic editor (optional)
 - <IP_name>.html
 - HTML generation report
 - Instantiation templates
- Located in <IP_name>/synth folder
 - <IP_name>.[v|vhd]
 - IP core top-level file connecting all components together
 - Submodule library folders for synthesis
 - Unique folder for each submodule (if any)
 - Combination of Verilog, SystemVerilog, and/or VHDL files representing submodule
- Files for simulation (optional)

IP Parameter Editor Output Files Example

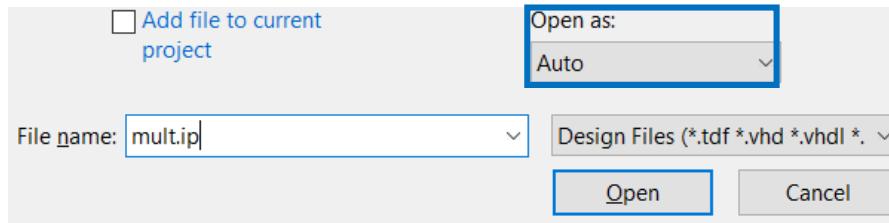


IP Parameter Editor Generate menu → Show Instantiation Template



Methods for Editing Existing IP Parameters

1. Double-click or right-click IP core in **IP Components** view of **Project Navigator**
2. Double-click in **Upgrade IP Components** dialog box
3. Open IP core's .ip file as **Auto**



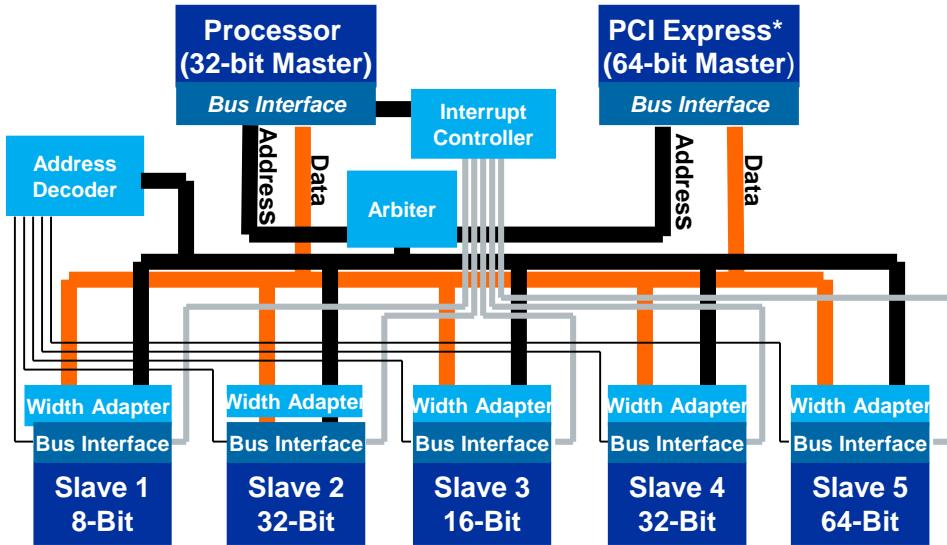
- Change parameters and regenerate IP core

System Design Entry With Platform Designer

- Simplifies complex system development
 - Automatic interconnect generation
- Raises the level of design abstraction
 - High level design and system visualization
- Provides a standard platform: IP integration, custom IP authoring, IP verification
- Enables design re-use
- Scales easily to meet the needs of end product
- Reduces time to market
 - Reduces design development time
 - Eases verification

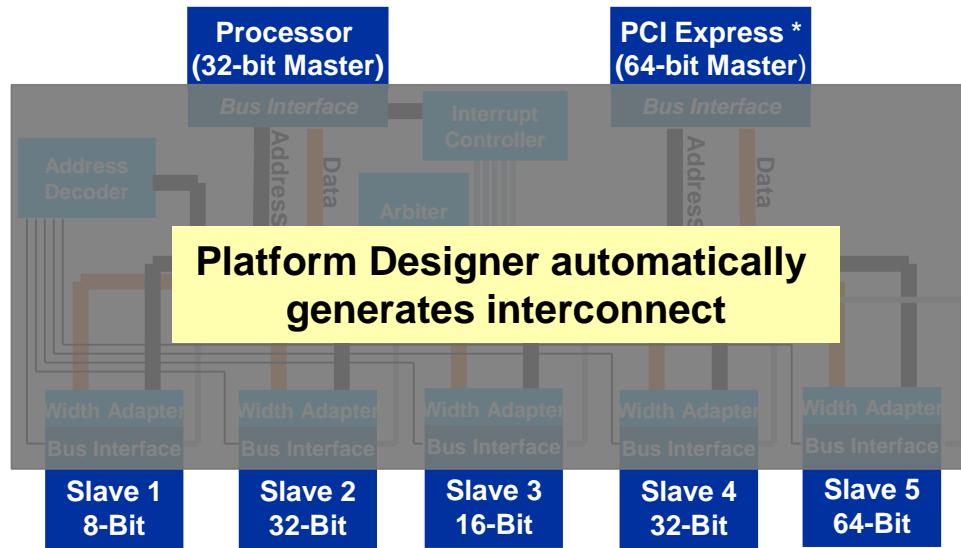
But What Exactly is Platform Designer?

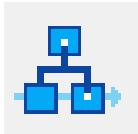
- Components in system use different interfaces to communicate (some standard, some non-standard)
- Typical system requires significant engineering work to design custom interface logic
- Integrating design blocks and intellectual property (IP) is tedious and error-prone



Automatic Interconnect Generation

- Avoids error-prone integration
- Saves development time with automatic logic & HDL generation
- Enables you to focus on value-add blocks
- Platform Designer improves productivity by automatically generating the system interconnect logic





The Platform Designer GUI

Access in Tools menu, toolbar, or Tasks window

The screenshot shows the Qsys Platform Designer interface with several tabs open:

- System Contents**: Displays a block diagram of the system. Components include a PLL, push_button_reg, av_sm_master, and led_out. Connectors like clk, pll, push_button_switches, sys_clk, and led_out are shown with their pin assignments.
- Hierarchy**: Shows the project structure with components like sm_transfer_system, clk, ext_mem_bus, greenled_out, hex0_out, hex1_out, hex2_out, and led_out.
- Messages**: Lists warnings and errors. Warnings include "sm_transfer_system.sram_controller Prop", "sm_transfer_system pll Able", and "sm_transfer_system pll pll.linking conduit". Errors include "sm_transfer_system.dma_source_to_ssram [Interrupt sender dma_source_to_ssram.irq is not connected to an interrupt receiver]".

A blue oval highlights the tabs at the top of the interface, and a blue arrow points to the "System Contents" tab.

Draggable, detachable tabs

Platform Designer

- Big idea: IP decoupled from system through .ip files
 - Allows for creation of *generic components* that define just the interface and signal requirements of the component
 - A “blackbox” or interface footprint
 - Redefine IP interfaces to fix issues
 - Good for component placeholders and team-based design
 - Good for revision control
- Support for IP_XACT (.ipxact) standard for delivery of IP from multiple vendors
 - <http://www.accelera.org/downloads/standards/ip-xact>
- Define or import system top-level interface requirements for connecting to rest of design

Platform Designer Resources (1)

- Platform Designer chapters in [Volume 1](#) of the Intel® Quartus® Prime Design Software Handbook
- Instructor-led training
 - [Introduction to the Platform Designer](#)
 - [Advanced Platform Designer Methodologies](#)

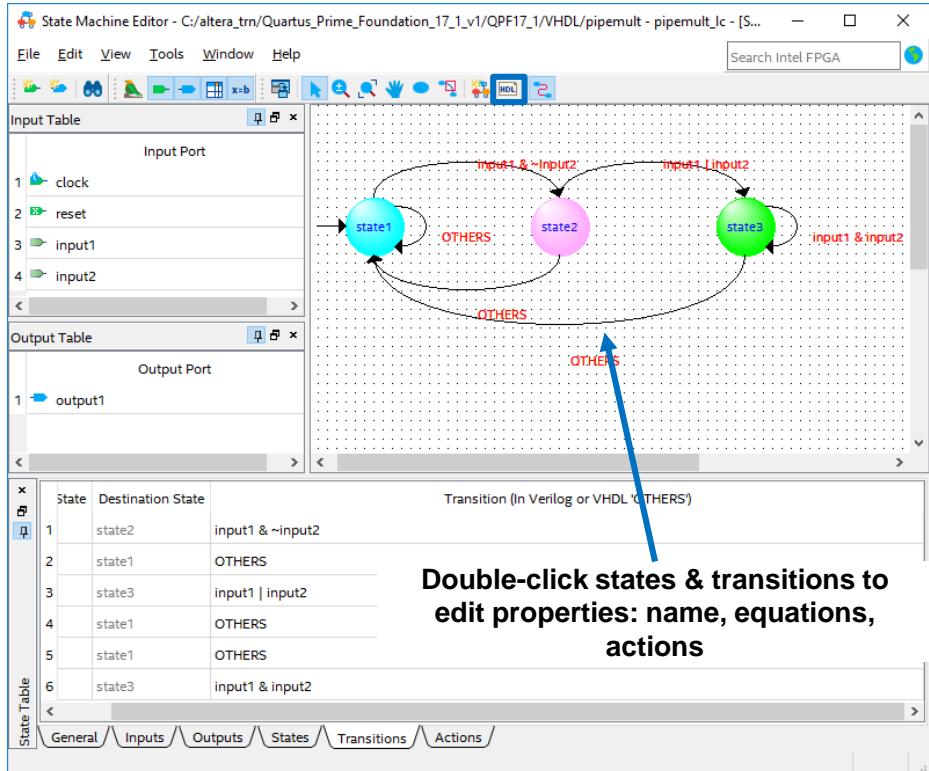
Platform Designer Resources (2)

- Online training
 - [Introduction to Platform Designer](#)
 - [Creating a System Design with Platform Designer](#)
 - Advanced System Design Using Platform Designer series
 - [Component & System Simulation](#)
 - [System Verification with System Console](#)
 - [Platform Designer System Optimization](#)
 - [Utilizing Hierarchy in Platform Designer Designs](#)
- Platform Designer
 - System Design with Platform Designer online training
 - <https://www.altera.com/support/training/course.html?courseCode=OQSYSPRO>
 - Introduction to Platform Designer chapter in Volume 1 of the Intel® Quartus® Prime Design Software Handbook

State Machine Editor

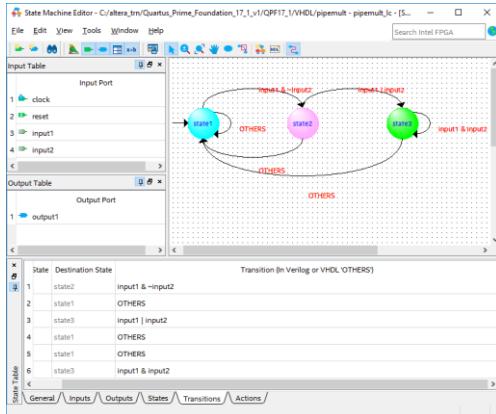
- Create state machines in GUI
 - Manually by adding individual states, transitions, and output actions
 - Automatically with State Machine Wizard (Tools menu & toolbar)
- Generate state machine HDL code (required)
 - VHDL
 - Verilog
 - SystemVerilog

File menu → New or Tasks window
Select *State Machine File (.smf)*



From .smf to HDL

- Generate optimized code (Verilog, SystemVerilog, or VHDL)
- Automatically added to project
- Required for use



```
Text Editor - C:/altera_trn/Quartus_Prime_Foundation_17_1_v1... File Edit View Project Tools Window Help Search Intel FPGA
module sm1 (
    input clock;
    input reset;
    input input1;
    input input2;
    output reg fstate;
    parameter state1=0,state2=1,state3=2;
);
begin
    if (clock) begin
        if (reset) begin
            fstate <= state1;
        end
        else begin
            case(fstate)
                state1: begin
                    if ((input1 & ~input2))
                        fstate <= state2;
                    else
                        fstate <= state1;
                end
                state2: begin
                    if ((input1 | input2))
                        fstate <= state3;
                    else
                        fstate <= state1;
                end
                state3: begin
                    if ((input1 & input2))
                        fstate <= state3;
                    else
                        fstate <= state1;
                end
                default: begin
                    $display ("Reach undefined state");
                end
            endcase
        end
    end
end
endmodule
```

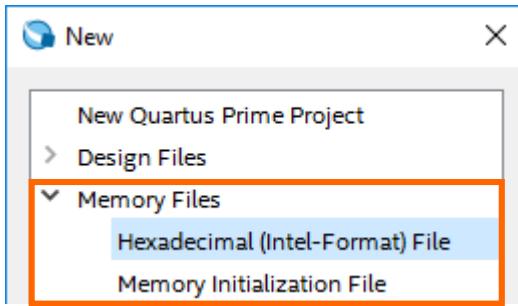
Memory Editor

- Create or edit memory initialization files
 - Intel HEX (**.hex**)
 - Preferred and compatible with 3rd-party tools
 - Intel® FPGA-specific (**.mif**) format
- Design entry
 - Initialization file data sent to device during device programming to initialize memory blocks
- Simulation
 - Use to initialize memory blocks before simulation or after breakpoints

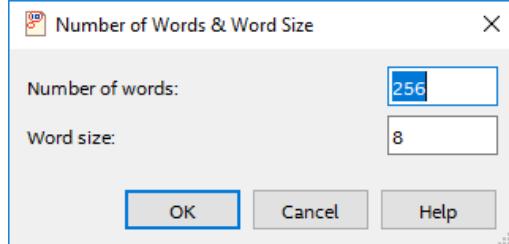
Create Memory Initialization File

File menu → New or Tasks window

1) HEX or MIF format



2) Select memory size



3) Memory space editor opens

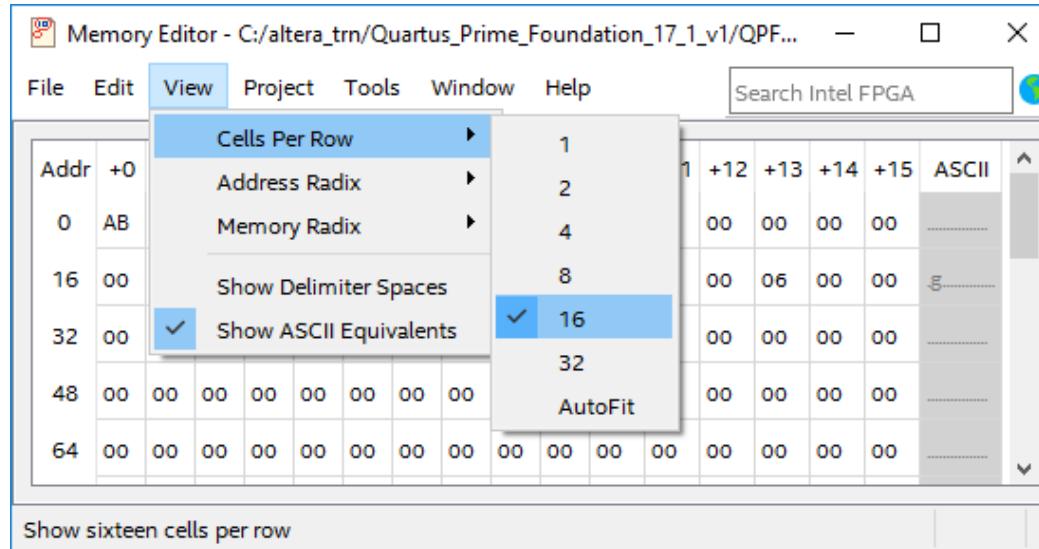
The screenshot shows the Memory Editor window displaying a memory dump. The table has columns for Address (Addr), Data bytes (+0 to +7), and ASCII representation. The data is as follows:

Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
0	AB	FF	DF	FF	01	00	02	00
8	03	00	04	00	00	00	00	00
16	00	67	00	00	00	05	05	00
24	00	00	00	00	00	06	00	00
32	00	00	08	00	00	00	00	00
40	00	00	00	00	00	00	00	00
48	00	00	00	00	00	00	00	00

Change Options

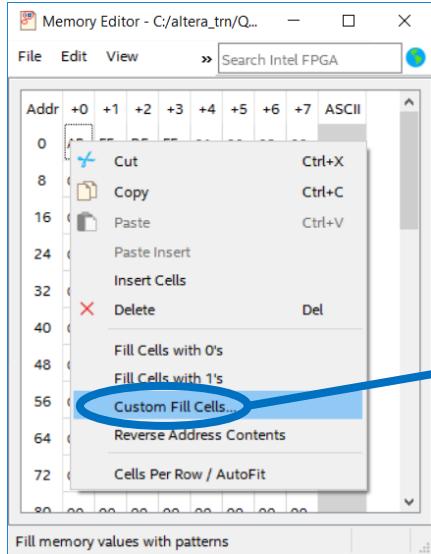
View options of memory editor

- **View menu → select from available options**

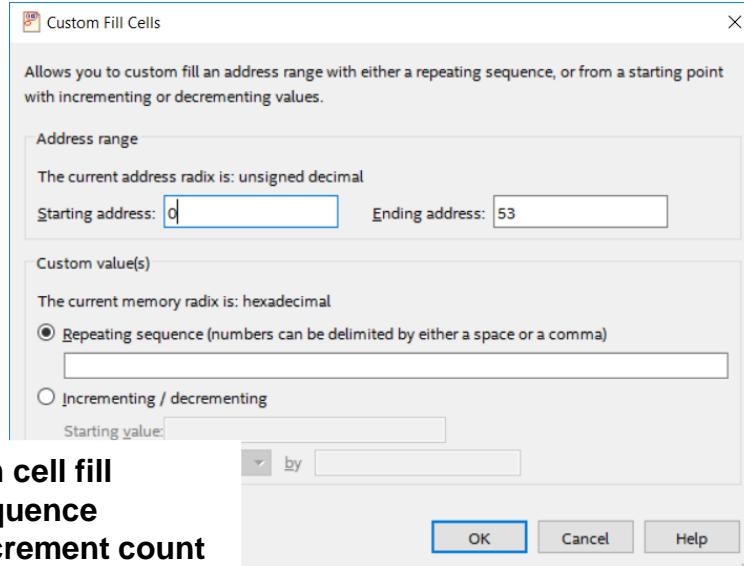


Edit Contents

- Edit contents of memory file
- Save memory file as .hex or .mif file



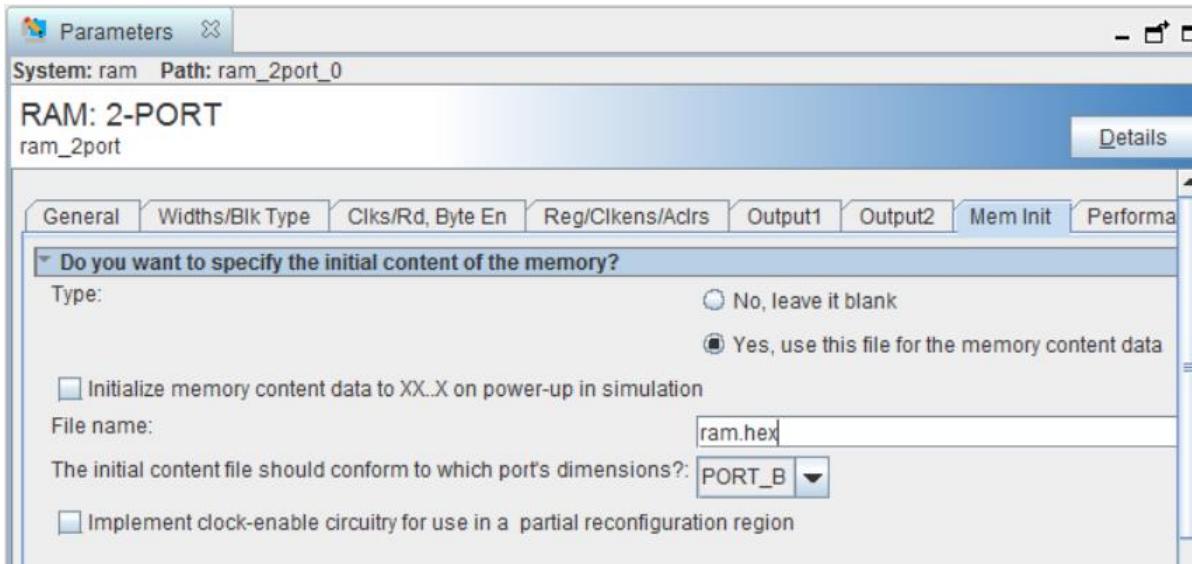
- Select address location & type in a value
- Select the address & right-click to select fill option from menu
- Copy & paste from spreadsheet



Specify custom cell fill
• Repeating sequence
• Increment/decrement count

Using Memory File in Design

Specify .mif or .hex file in IP Parameter Editor

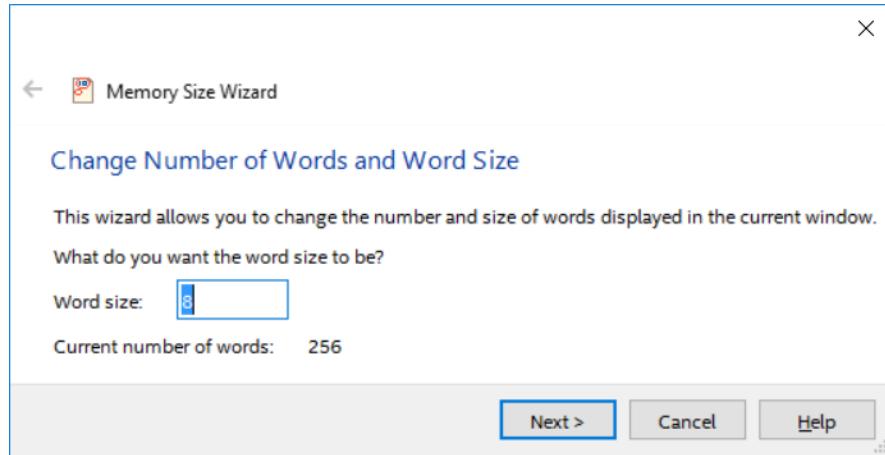


**May also specify .mif file in HDL using the
ram_init_file synthesis attribute**

Memory Size Wizard

Allows editing the size of already open memory file

- Use the Memory Size Wizard (**Edit** menu)
 - Edit word size
 - Edit number of words
 - Specify how to handle word size change
 - Pad words
 - Combine words
 - Truncate words
 - Add words



Importing 3rd-Party EDA Tool Files

- Interface with industry-standard EDA tools that generate netlist files
 - EDIF 2 0 0 (**.edf**)
 - Verilog Intel® Quartus® Prime software Mapping (**.vqm**)
- To import and use netlist files
 - Specify EDA tool in the Intel® Quartus® Prime software settings
 - Instantiate block(s) in design
 - Add **.edf/.vqm** file(s) to Intel Quartus Primed software project

3rd-Party Synthesis Tool Support

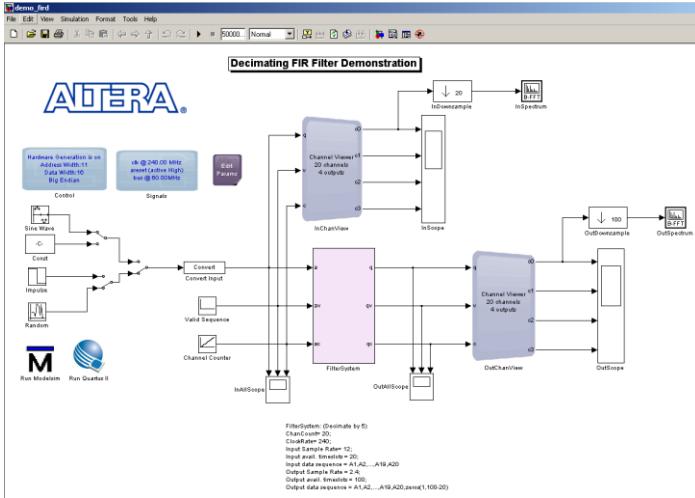
- Precision* RTL
- Precision RTL Plus
- Synplify*
- Synplify Pro
- Synplify Premier



System Design Entry Tools

- Platform Designer system development tool (*discussed earlier*)
- DSP Builder for Intel® FPGA Standard/Advanced Blocksets
- Intel® FPGA SDK for OpenCL* Compiler

DSP Builder for Intel® FPGAs (DSPB) Standard/Advanced Blocksets



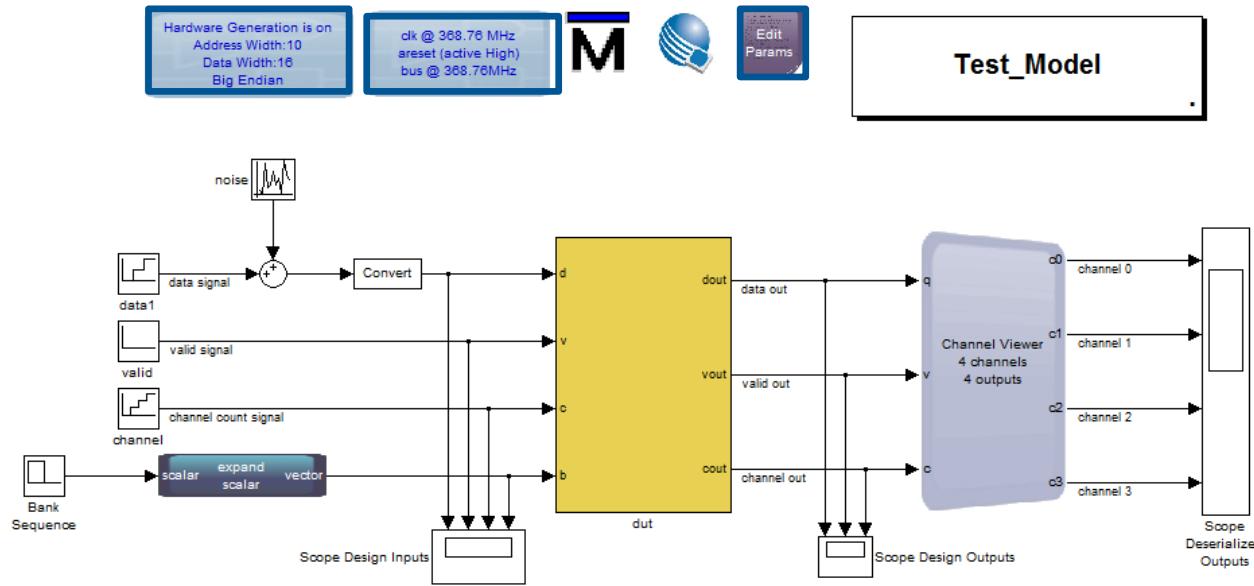
```
add_cost <= resize(Mux_out1(0), 32);
add_cost_1 <= resize(Mux_out1(1), 32);
add_temp <= resize(add_cost_33) + resize(add_cost_1, 33);
sum_1 <= (31 => '0', OTHERS => '1') WHEN add_temp(32) = '0' AND add_temp(31) /= '0'
ELSE (31 => '1', OTHERS => '0') WHEN add_temp(32) = '1' AND add_temp(31) /= '1'
ELSE (add_temp(31 DOWNTO 0));
add_cost_2 <= sum_1;
add_cost_3 <= resize(Mux_out1(2), 32);
add_temp_1 <= resize(add_cost_2, 33) + resize(add_cost_3, 33);
sum_2 <= (31 => '0', OTHERS => '1') WHEN add_temp_1(32) = '0' AND add_temp_1(31) /= '0'
ELSE (31 => '1', OTHERS => '0') WHEN add_temp_1(32) = '1' AND add_temp_1(31) /= '1'
ELSE (add_temp_1(31 DOWNTO 0));
add_cost_4 <= sum_2;
add_cost_5 <= resize(Mux_out1(3), 32);
add_temp_2 <= resize(add_cost_4, 33) + resize(add_cost_5, 33);
sum_3 <= (31 => '0', OTHERS => '1') WHEN add_temp_2(32) = '0' AND add_temp_2(31) /= '0'
ELSE (31 => '1', OTHERS => '0') WHEN add_temp_2(32) = '1' AND add_temp_2(31) /= '1'
ELSE (add_temp_2(31 DOWNTO 0));
add_cost_6 <= sum_3;
add_cost_7 <= resize(Mux_out1(4), 32);
add_temp_3 <= resize(add_cost_6, 33) + resize(add_cost_7, 33);
Sum_out1Tmp <= (31 => '0', OTHERS => '1') WHEN add_temp_3(32) = '0' AND add_temp_3(31) /= '0'
ELSE (31 => '1', OTHERS => '0') WHEN add_temp_3(32) = '1' AND add_temp_3(31) /= '1'
ELSE (add_temp_3(31 DOWNTO 0));
Sum_out1 <= (12 => '0', OTHERS => '1') WHEN Sum_out1Tmp(31) = '0' AND Sum_out1Tmp(30 DOWNTO 12) /= "00000000000000000000000000000000"
ELSE (12 => '1', OTHERS => '0') WHEN Sum_out1Tmp(31) = '1' AND Sum_out1Tmp(30 DOWNTO 12) /= "11111111111111111111111111111111"
ELSE (Sum_out1Tmp(12 DOWNTO 0));
Out1 <= std_logic_vector(Sum_out1);
```

HDL automatically
optimized for system
clock frequency & latency



DSP Builder for Intel® FPGAs Advanced Blockset Example

Top-level of a DSP Builder for Intel® FPGAs design is a testbench



DSP Builder for Intel® FPGAs Design Standard and Advanced Blocksets (1)

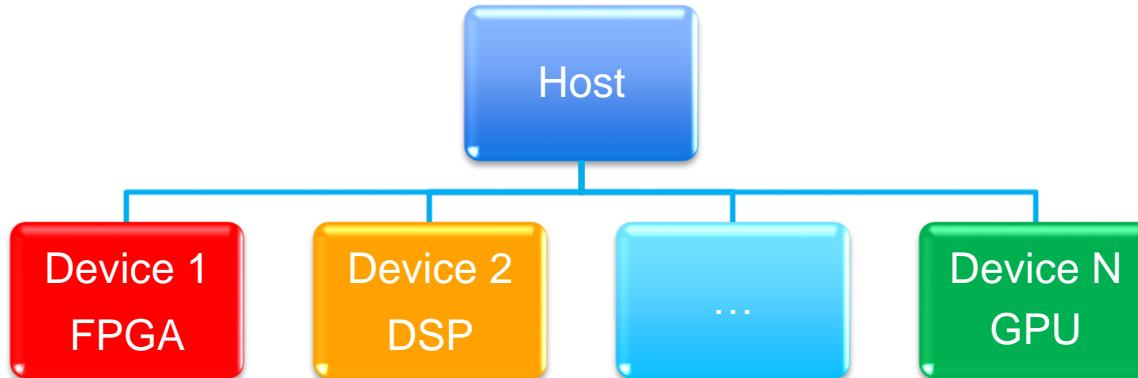
- DSP Builder for Intel® FPGA Standard Blockset
 - Cycle accurate behavior (works the way you designed it)
 - Multiple clock domain design
 - Control rich with state machine or backpressure
 - Access detailed device features
 - Custom HDL code import support
 - Hardware-in-the-loop simulation support

DSP Builder for Intel® FPGAs Design Standard and Advanced Blocksets (2)

- DSP Builder for Intel® FPGAs Advanced Blockset
 - Specification driven design with automatic pipeline and folding
 - Multichannel designs with automatic vectorized inputs
 - Single system clock for the main datapath logic
 - Design exploration
- DSP solutions web site
 - <https://www.altera.com/solutions/technology/dsp/overview.html>

Open Computing Language (OpenCL*)

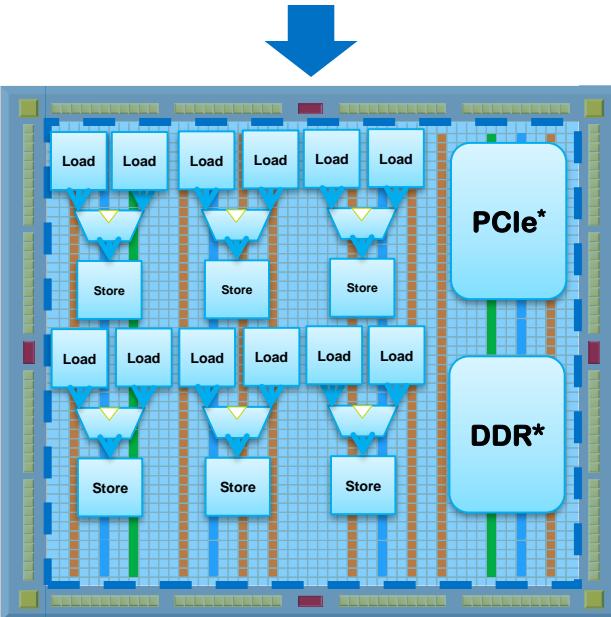
- Framework for heterogeneous programs
- Royalty-free
- Allows general purpose programs to run on multiple platforms
- Adaptable to obtain high performance



Intel® FPGA SDK for OpenCL*

- Translates OpenCL* source file into custom-generated hardware image to be loaded onto FPGA
- Provides API for use by OpenCL host application to execute and communicate with hardware image
- Intel® FPGA SDK for OpenCL
 - <https://www.altera.com/products/design-software/embedded-software-developers/opencl/overview.html>

```
// kernel.cl
__kernel void KernelName(...)
{
    int i = get_global_id(0);
    c[i] = a[i] + b[i];
}
```



Exercise 2

Design Entry Summary

- Multiple design entry methods supported
 - Text (Verilog, VHDL, State Machine Editor output)
 - 3rd-party netlist (VQM, EDIF)
 - Schematic
 - System (Platform Designer)
- IP Parameter Editor parameterizes IP cores
- Memory Editor allows generation of memory initialization files
- 3rd-party EDA tools supported for design entry & synthesis

Design Entry Support Resources (1)

- Intel® Quartus® Prime Design Software Handbook chapters ([Volume 1](#))
 - Recommended Design Practices
 - Recommended HDL Coding Styles
 - Platform Designer chapters
 - Creating a System With Platform Designer
 - Creating Platform Designer Components
 - Platform Designer Interconnect
 - 3rd-party EDA tool chapters

Design Entry Support Resources (2)

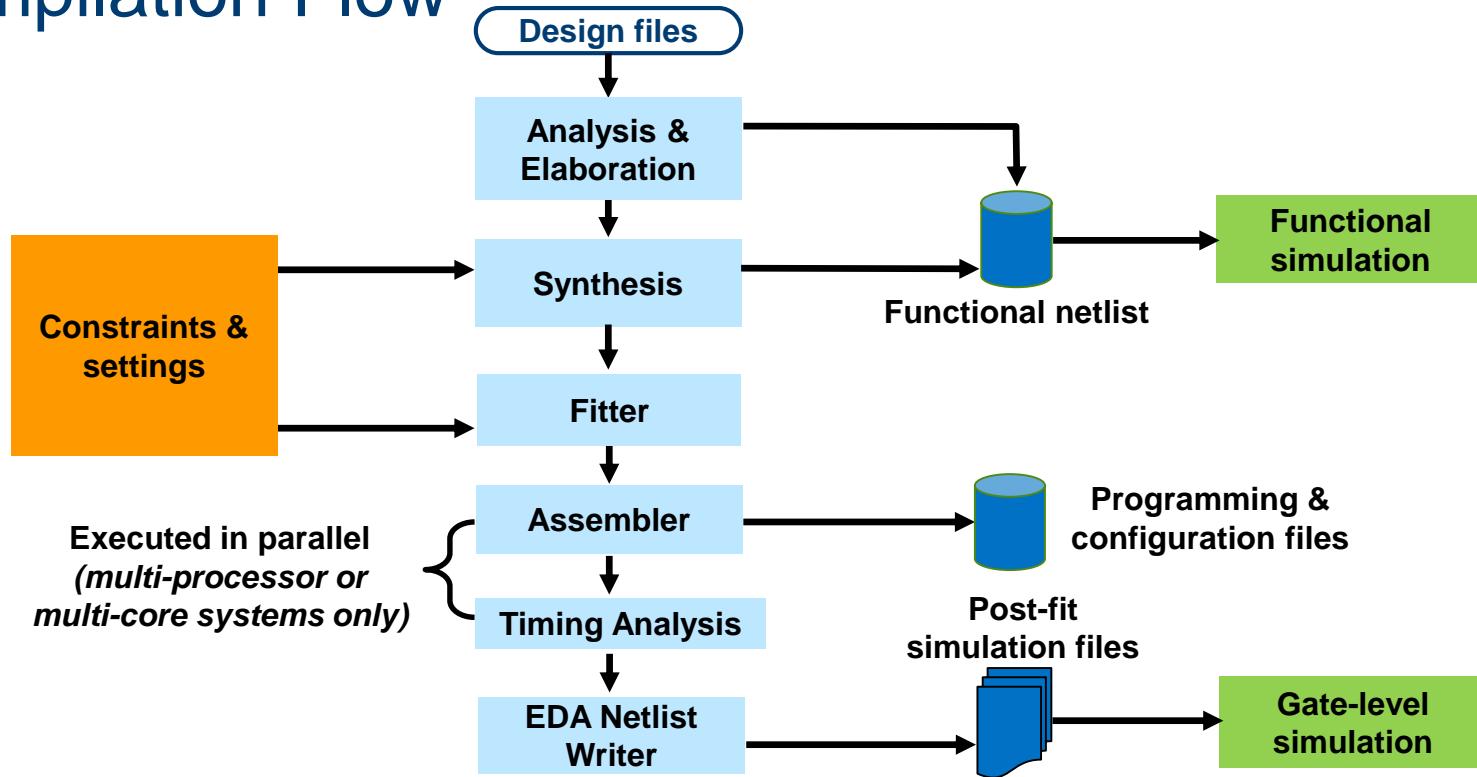
- Training courses & demonstrations
 - [VHDL & Verilog HDL Basics](#) (online courses)
 - [Introduction to Verilog HDL & Advanced Verilog HDL Design Techniques](#) courses
 - [Introduction to VHDL & Advanced VHDL Design Techniques](#) courses
 - [SystemVerilog with the Intel® Quartus® Prime Design Software](#) (online)
 - [Designing with the DSP Builder for Intel® FPGAs Advanced Blockset](#)
 - [Introduction to OpenCL* for Intel® FPGAs](#)
 - [Introduction to Parallel Computing with OpenCL on FPGAs](#)



The Intel® Quartus® Prime Design Software: Foundation

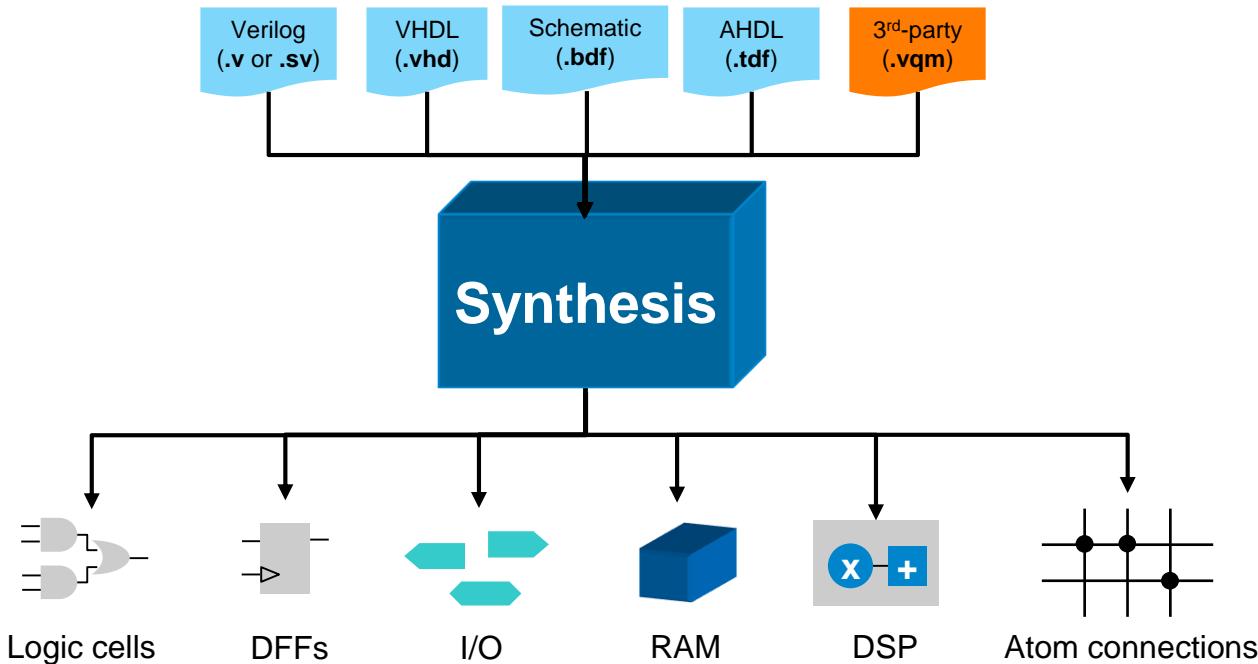
Intel Quartus Prime Software Compilation

Intel® Quartus® Prime Design Software Full Compilation Flow



What is Synthesis?

Translates HDL source files into an **atom netlist**



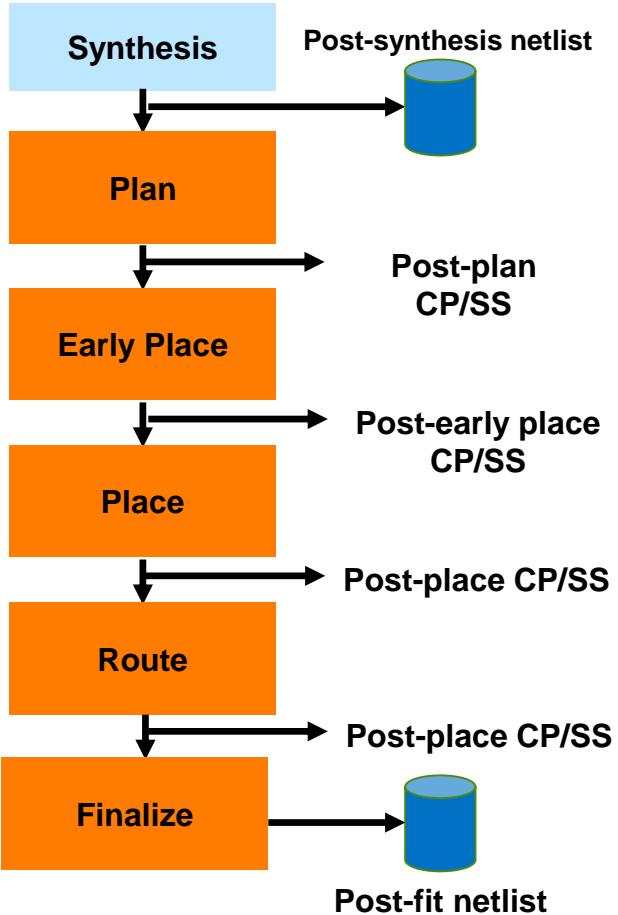
Intel® Quartus® Prime Pro Synthesis

- New synthesis engine found only in the Intel® Quartus® Prime Pro Edition Design Software
- New front-end language parser
 - Improved support for all IEEE RTL languages
 - Stricter syntax/semantic checks
- New algorithms and truly parallel synthesis
- Generates an advanced hierarchical database compatible with all editions of the Intel Quartus Prime software
- Targets up to 16 processors
- *Using Spectra-Q Synthesis in the Quartus® Prime Software* online training
 - <https://www.altera.com/support/training/course.html?courseCode=OSYNQPRO>
- *Design Synthesis* section of *Design Compilation* chapter in Volume 1 of the Intel Quartus Prime Software Pro Handbook

What is the Fitter?

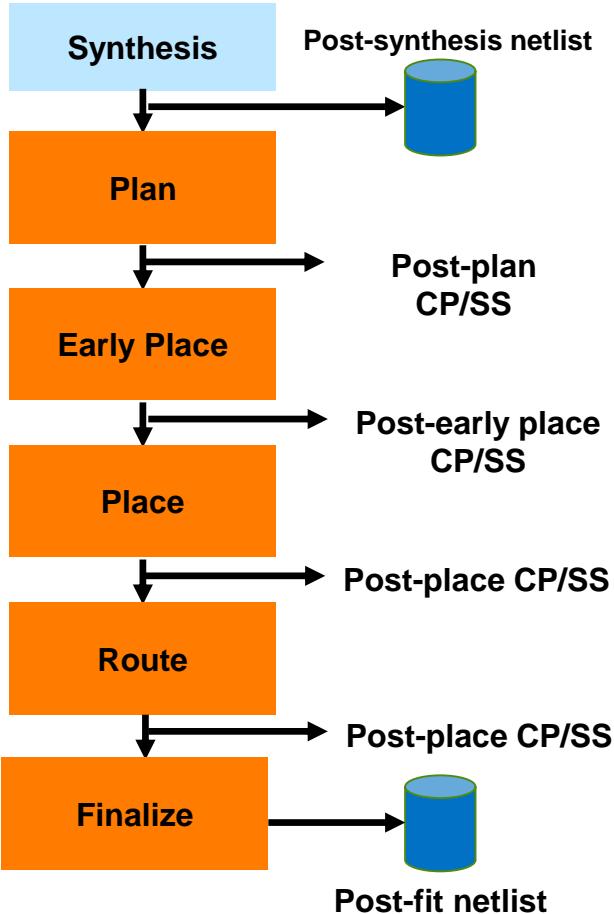
Advanced place & route engine for finding a valid “solution” in a “reasonable” amount of time

- Consists of 4 stages
- Stages can be all run or individually
 - End of running each stage referred to as a **checkpoint (CP)**
- Prior stages must be complete before running later stages



What is the Fitter? (2)

- Output of each stage is a database referred to as a **snapshot (SS)**
- Useful for incremental optimization
 - Optimize design at each checkpoint
 - Save compile time by only running affected stage(s)



Fitter Stages

Plan

- Periphery (I/O) placement and routing, clock resource selection

Early Place

- Early assignment of core logic to device resources
- More pessimistic results within 20% of final routed results

Place

- Core resource placement (logic elements, registers, DSP, RAM)

Fitter Stages (2)

Route

- Core routing connections made

Finalize

- Post-routing optimizations

Processing Menu, Toolbar, and Command Line Options (1)

- **Start Compilation (Full)**
- **Start Analysis & Elaboration**
 - Checks syntax & builds hierarchy
 - Performs initial synthesis
- **Start Analysis & Synthesis**
 - Synthesizes & optimizes code
- **Start Fitter**
 - Places & routes design
 - Generates output netlists

The screenshot shows the Quartus software interface. On the left, the Processing menu is open, displaying various compilation and analysis options. To the right of the menu, there is a toolbar with icons for STOP, Start, and other tools. Below the menu, a command-line reference table maps software commands to their terminal counterparts and keyboard shortcuts.

Quartus Software Command	Terminal Command	Keyboard Shortcut
Start	quartus_start	
Update Memory Initialization File	quartus_update_mif	
Compilation Report	quartus_report	Ctrl+R
Compilation Dashboard	quartus_dashboard	Ctrl+Shift+R
Power Analyzer Tool	quartus_power_analyzer	
Receive Compilation Status Notifications	quartus_receive_notifications	
Start Analysis & Elaboration	quartus_syn	
Start Analysis & Synthesis	quartus_fit	
Start Fitter	quartus_asm	
Start Fitter (Plan)		
Start Fitter (Early Place)		
Start Fitter (Place)		
Start Fitter (Route)		
Start Fitter (Finalize)		
Start Assembler		
Start TimeQuest Timing Analyzer	quartus_timequest	Ctrl+Shift+T
Start EDA Netlist Writer	quartus_edanetlist	
Start Power Analyzer	quartus_power_analyzer	Ctrl+Shift+P
Start Rapid Recompile	quartus_rapid_recompile	
Start Test Bench Template Writer	quartus_tb_template	

Processing Menu, Toolbar, and Command Line Options (2)

- **Start Timing Analyzer**

- **Start Assembler**

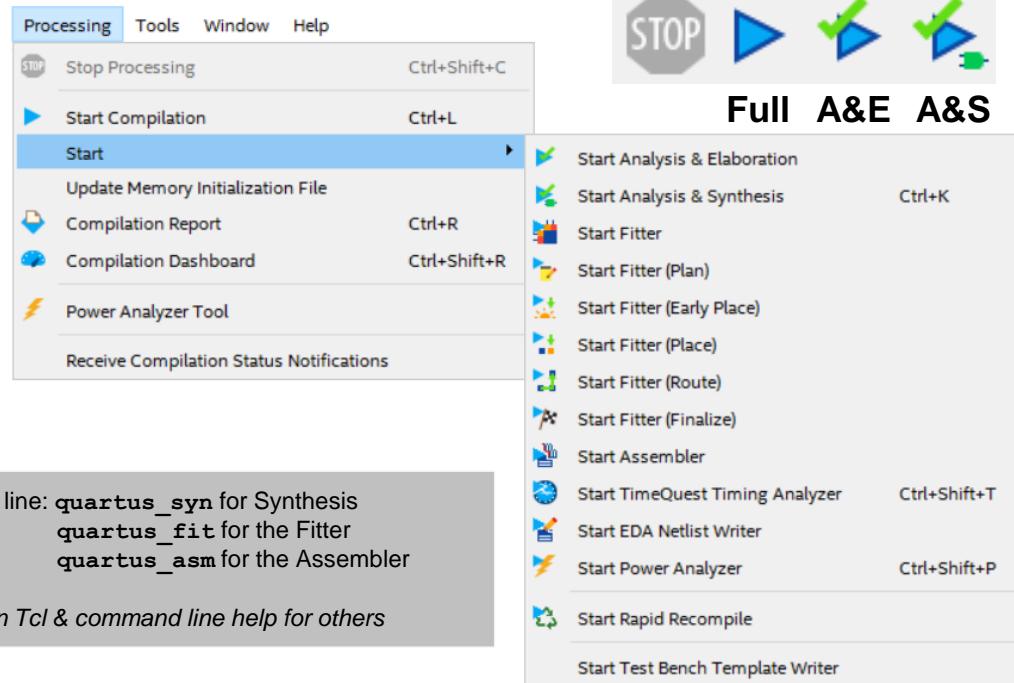
- Generate programming files

- **Start Rapid Recompile**

- Run compilation preserving previous unmodified results

- **Start Power Analyzer**

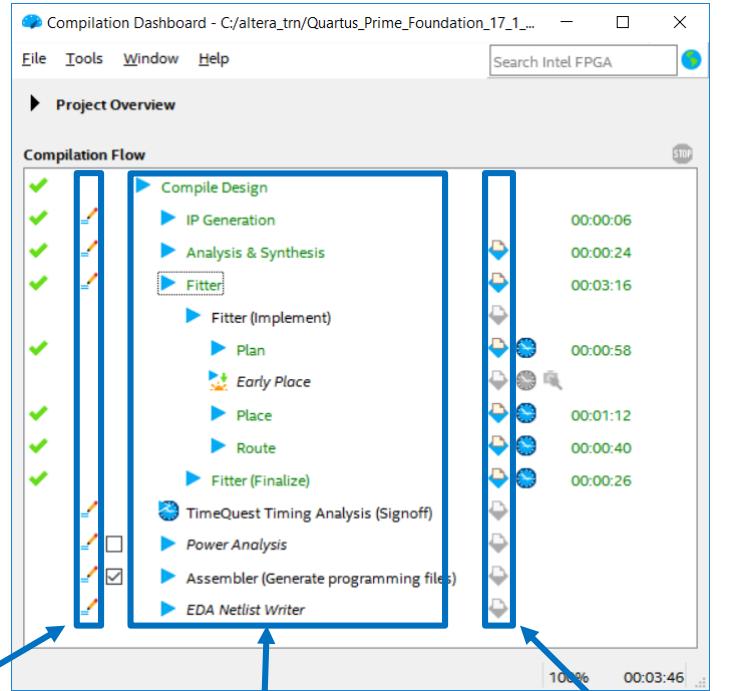
- Perform an early power estimate



Compilation Dashboard

- Control execution of all compilation processes
- Run individual Fitter stages
- Quickly adjust settings or view compilation report for completed stages
- Choose **Compilation Flow**
 - **Compilation:** run full compilation (all processes) or main compilation processes individually
 - **Per-Stage Compilation:** run Fitter stages or other main compilation processes individually

Processing menu or Tasks window



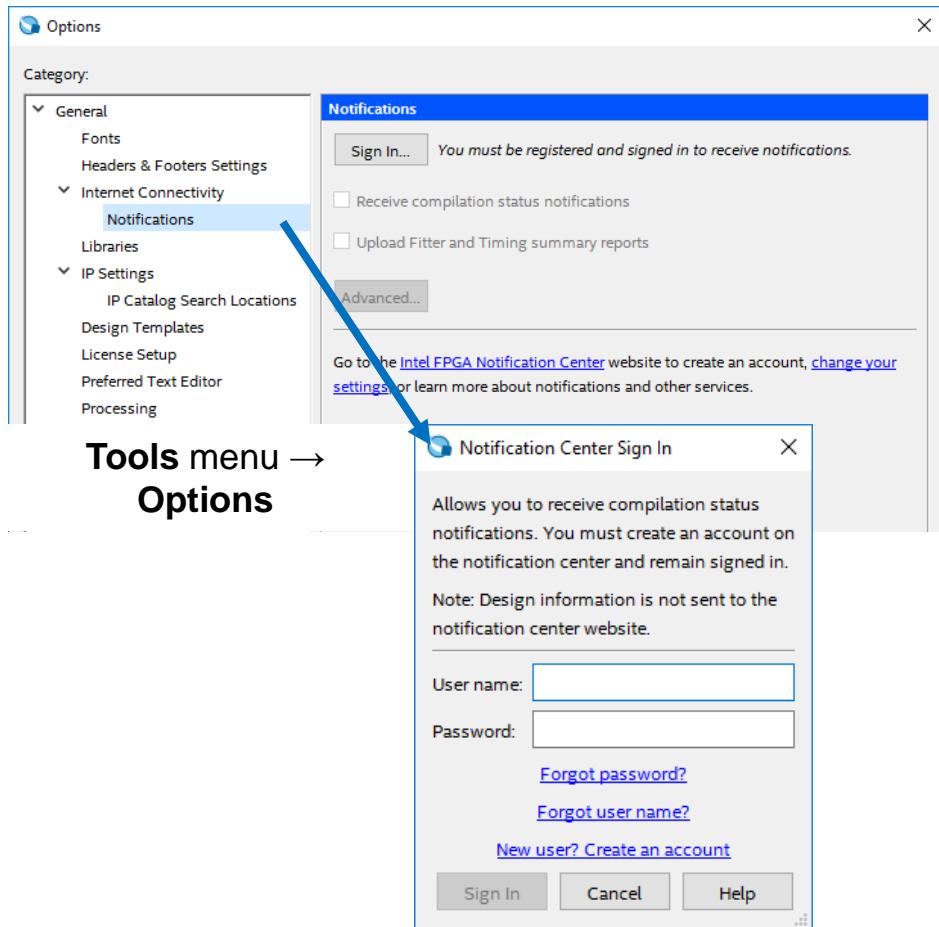
Edit settings
for process

Click button or double-
click process to run

View Compilation
Report for process

Notification Center

- Monitor compilations via the web
- Receive email notifications when compilations finish
- Compile is local only; no private design data or IP is transferred
- Receive error or critical warning messages (optional)



Using the Notification Center

- Sign up for Intel® FPGA web account
<https://cloud.altera.com>
 - Requires valid email address
- Connect Intel® Quartus® Prime Design Software installation to web account by signing in
- Compile design from command-line or GUI
 - Compile status (only) reflects on web
 - <https://cloud.altera.com/notify> shows list of running or finished compiles
- Receive email notifications and web links summarizing compile status
- Delete records in web account at anytime

The screenshot shows the Intel FPGA web interface. At the top, there's a navigation bar with the Intel logo and a 'Login' button. Below it is a 'Welcome to the Quartus Prime software online extension' message. To the right is a 'Sign in' form with fields for 'Username or Email' and 'Password', along with 'Forgot password?' and 'Forgot username?' links, and a 'Sign in' button. In the center, there are three icons: 'Find the best device for your design' (a chip icon), 'Receive notifications' (a bell icon), and 'Download design templates' (a shopping cart icon). Below these is a note about an early access beta. At the bottom, there's a table titled 'Project name' showing a single compilation named 'pipemult'. An orange arrow points to the 'Single Compile' button in the table header. To the right, text says 'Click on compilation listed to open report'. A blue arrow points to the status column in the table, which shows 'Running at Analysis & Synthesis (89%)'.

Notification Center Report

The screenshot shows a detailed report for the project "pipemult".

Details:

Host	selzinga-MOBL
Compile Status	Successful 100%
Owner	selzinga
Software Version	Quartus II 17.1
Started On	10/31/2017 8:24 a.m.
Completed On	10/31/2017 8:30 a.m.
Duration	5m 53s

Stages:

Flow	Runtime	Status
IP Generation Tool	4s	Done
Synthesis	3s	Done
Analysis & Synthesis	25s	Done
Fitter	3m 35s	Done
TimeQuest Timing Analyzer	14s	Done
Timing Analysis (Finalize)	11s	Done
Assembler	1m 8s	Done
QOR Processor	13s	Done

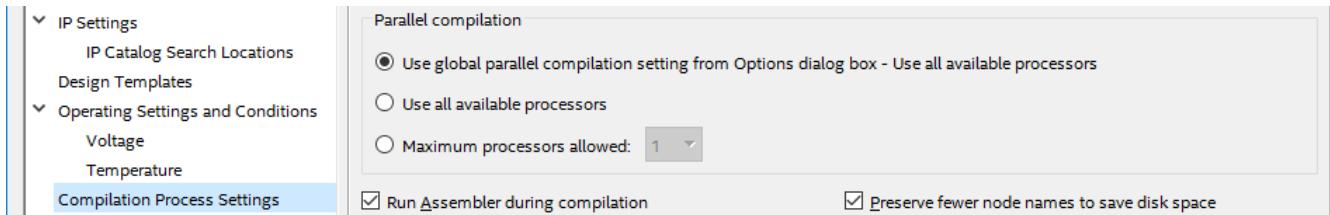
Reports:

Filter: Slack

Top-level Entity Name	pipemult
Total block memory bits	2,048 / 55,562,240 (< 1 %)
Total PLLs	0 / 144 (0 %)
Logic utilization (in ALMs)	0 / 427,200 (0 %)
Total registers	32
Total virtual pins	0
Quartus Prime Version	17.1.0 Build 237 10/04/2017 SJ Pro Edition

Reducing Compile Times

- Running individual compilation processes
- Parallel compilation
 - Take advantage of multi-processor, multi-core machines



**Assignments menu → Settings
(discussed in more detail later)**

Rapid Recompilation



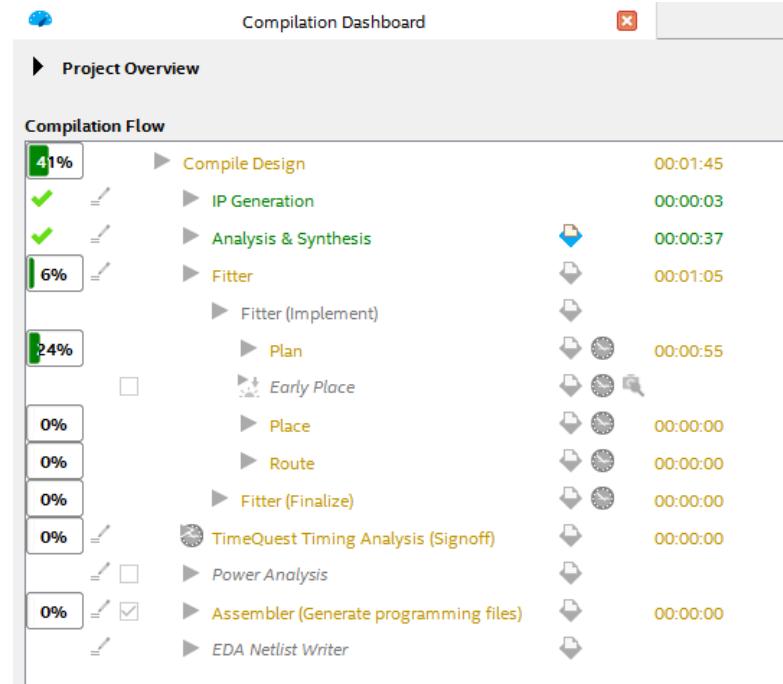
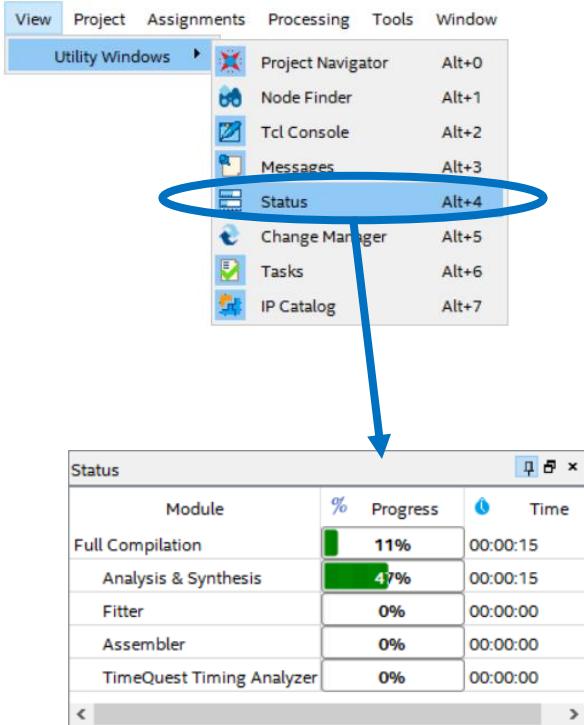
- Reuse previous compilation results to update design for *small* changes (affect less than 5% of design logic)
- Design changes supported by rapid recompilation
 - Node tapping with the Signal Tap embedded logic analyzer
 - Modify simple combinational logic
 - Modify state machine logic
 - Add pipeline registers
 - See Intel® Quartus® Prime Design Software Handbook for more
- “Disengages” if compiler determines netlist can’t be preserved

Processing menu → Start →
Start Rapid Recompile

Rapid Recompile Preservation Summary		
<<Filter>>		
	Type	Achieved
1	Placement (by node)	100.00 % (224 / 224)
2	Routing (by connection)	100.00 % (642 / 642)

**Rapid Recompile Preservation Summary
(Fitter folder of Compilation Report)**

Status Window and Compilation Dashboard



Messages Window

Message window displays informational, warning, and error messages

Filter messages by type

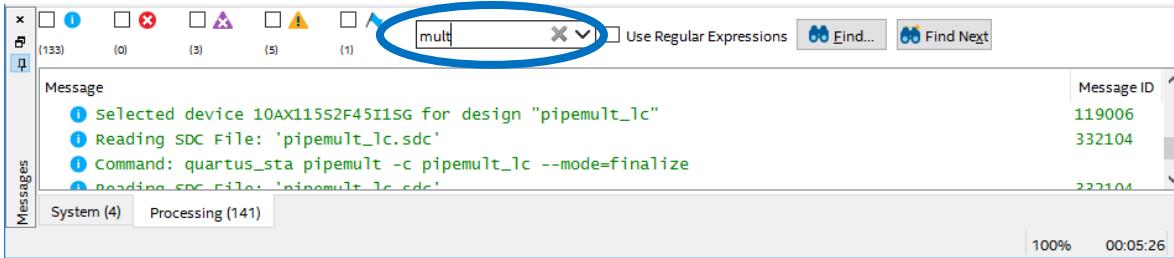
Info Error Critical Warning Flagged

Right-click to flag selected messages for later review

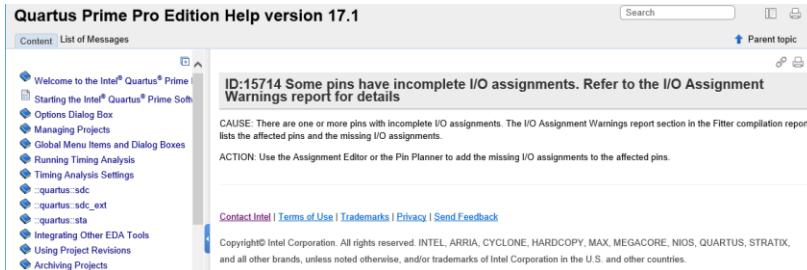
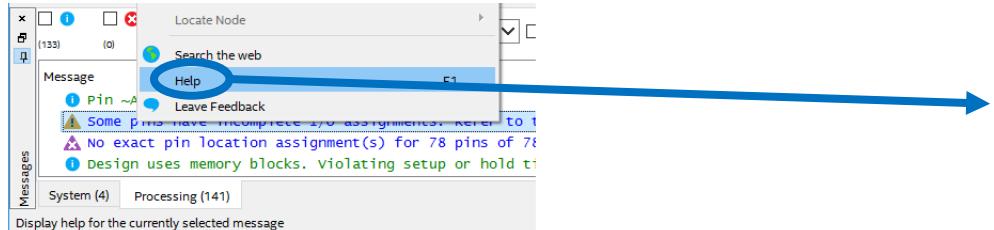
The screenshot shows the 'Messages' window interface. At the top, there is a toolbar with filter buttons for 'Info' (blue), 'Error' (red), 'Critical Warning' (purple), and 'Flagged' (blue). Below the toolbar is a status bar displaying message counts: (133) for Info, (0) for Error, (3) for Critical Warning, (5) for Flagged, and (1) for All. To the right of the status bar are buttons for 'Find...' and 'Find Next'. A dropdown menu is open, showing options: View, Project, Assignments, Proc, Project Navigator (Alt+0), Node Finder (Alt+1), Tcl Console (Alt+2), **Messages (Alt+3)** (which is circled in blue), Tasks, and IP Catalog. The main area of the window is titled 'Message' and contains a list of messages. The first message is a yellow warning: 'Some pins have incomplete I/O assignments. Refer to the I/O Assignment warnings report for details 15714'. The second message is a purple warning: 'No exact pin location assignment(s) for 78 pins of 78 total pins. For the list of pins please ref... 12677'. The third message is a blue info message: 'Design uses memory blocks. violating setup or hold times of memory block address registers for ei... 176045'. The fourth message is a green info message: 'plan updated with currently enabled project assignments 16210'. At the bottom of the window, there are tabs for 'System (4)' and 'Processing (141)'.

Message Searching & Help

- Search for keywords in the Messages window



- Right-click to search built-in help



Message IDs

- Benefits

- More easily reference messages in service requests
- Search for collateral documentation (self-help)
- Disable specific info or warning messages globally (`MESSAGE_DISABLE` assignment)

- Benefits for IP developers

- Disable info or warning messages specifically for an IP core
- Ship “warning-free” IP

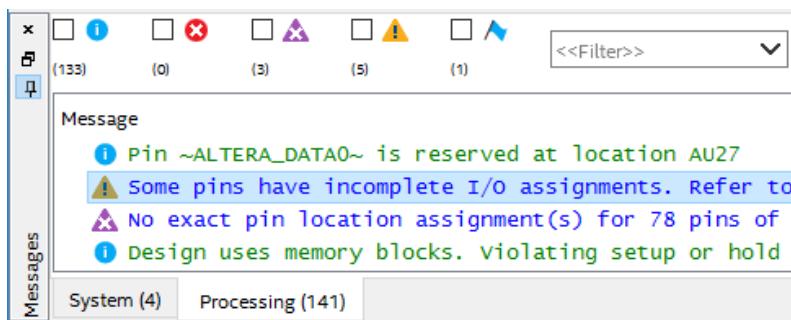
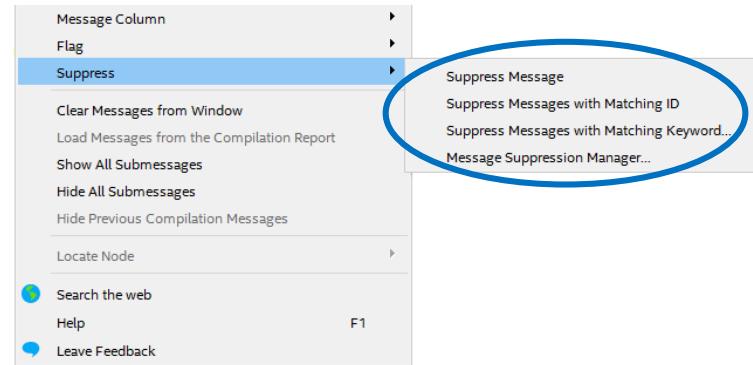
The screenshot shows a software interface with a toolbar at the top containing icons for information, error, warning, and other message types, along with a filter bar and search functions. Below this is a main pane titled 'Messages' which displays a list of messages. The messages are color-coded: green for informational, blue for warnings, and purple for errors. One specific message is highlighted in blue and circled in red on the right side of the screen. This circled message is about incomplete I/O assignments and has a 'Message ID' of 15714. Other visible message IDs include 12627, 12677, and 176045. At the bottom, there are tabs for 'System (4)' and 'Processing (141)'.

Message Suppression

Hide messages from current and future compiles

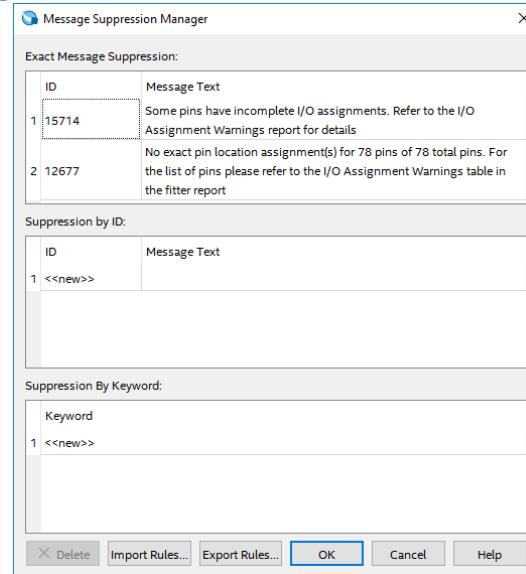
- E.g. known synthesis warning message already investigated
- Store suppression rules in *<revision_name>.srf* file

Right-click message to suppress exact message, messages with matching ID, or messages with matching keyword



Message Suppression Manager Tool

- View/edit/remove suppression rules
- Import and export message suppression rules



Viewing Compilation Results

- Intel® Quartus® Prime Design Software graphical tools available for
 - Understanding design processing
 - Verifying correct design results
 - Debugging incorrect results
- Compilation Report
- Viewers
 - RTL Viewer
 - Technology Map Viewer
 - State Machine Viewer
- Chip Planner

Compilation Report

- Contains all compilation processing information
 - Resource usage
 - Device pin-out
 - Settings and constraints applied
 - Messages
- Recommendation: Go through report for a design to get sense of information being provided
- Information also available as text files in **output_files** folder in project directory: *<revision_name>.syn.rpt*, *<revision_name>.fit.rpt*, *<revision_name>.fit.plan.rpt*, etc.

Compilation Report GUI



- Access from **Processing** menu, toolbar, or Compilation Dashboard

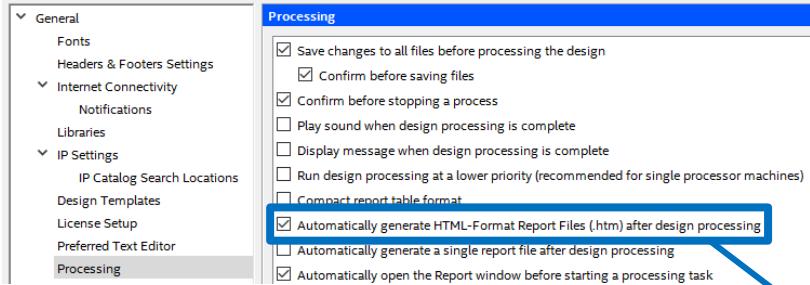
Each compiler process generates separate folder

The screenshot shows the 'Flow Summary' section of the Quartus Prime Compilation Report. The table provides detailed statistics for the compilation process:

Flow Summary	
Flow Status	Successful - Tue Sep 12 12:44:26 2017
Quartus Prime Version	17.1.0 Internal Build 169 08/23/2017 SJ Pro Edition
Revision Name	pipemult_lc
Top-level Entity Name	pipemult
Family	Arria 10
Device	10AX115S2F45I1SG
Timing Models	Final
Logic utilization (in ALMs)	116 / 427,200 (< 1 %)
Total registers	96
Total pins	78 / 960 (8 %)
Total virtual pins	0
Total block memory bits	2,048 / 55,562,240 (< 1 %)
Total DSP Blocks	0 / 1,518 (0 %)
Total HSSI RX channels	0 / 72 (0 %)
Total HSSI TX channels	0 / 72 (0 %)
Total PLLs	0 / 144 (0 %)

Compilation Report: HTML version

Tools menu → Options

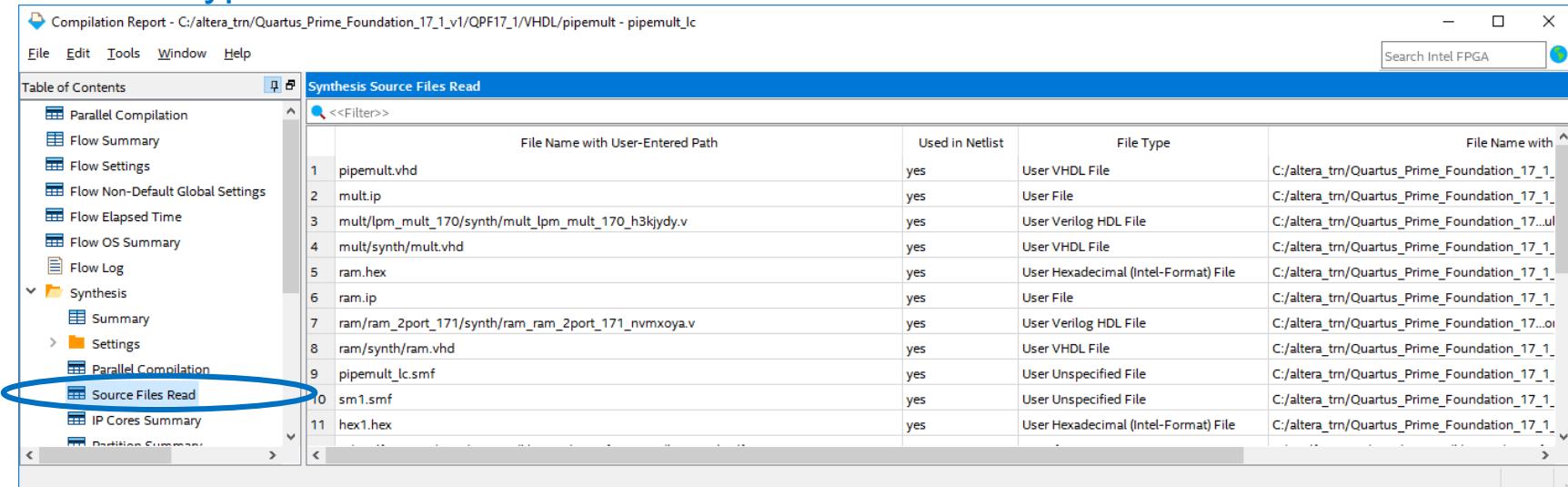


output_files folder

Table of Contents		Fitter Summary
Legal Notice		Fitter Status Successful - Tue Sep 12 13:08:43 2017
Fitter		Quartus Prime Version 17.1.0 Internal Build 169 08/23/2017 SJ Pro Edition
 Fitter Summary		Revision Name pipemult_lc
 Fitter Settings		Top-level Entity Name pipemult
 Parallel Compilation		Family Arria 10
 Fitter Partition Summary		Device 10AX115S2F45I1SG
 Fitter Netlist Optimizations		Timing Models Final
 Plan Stage		Logic utilization (in ALMs) 116 / 427,200 (< 1 %)
 Place Stage		Total registers 96
 Route Stage		Total pins 78 / 960 (8 %)
 Finalize Stage		Total virtual pins 0
 Fitter Messages		Total block memory bits 2,048 / 55,562,240 (< 1 %)
		Total RAM Blocks 1 / 2,713 (< 1 %)
		Total DSP Blocks 0 / 1,518 (0 %)
		Total HSSI RX channels 0 / 72 (0 %)
		Total HSSI TX channels 0 / 72 (0 %)
		Total PLLs 0 / 144 (0 %)

Example: Synthesis Source Files Read

List of all design files (user-coded & library) used during last compilation along with files' type and location



The screenshot shows the 'Compilation Report - C:/altera_trn/Quartus_Prime_Foundation_17_1_v1/QPF17_1/VHDL/pipemult - pipemult_lc' window. The 'Table of Contents' on the left lists various reports, with 'Source Files Read' highlighted by a blue oval. The main pane displays a table titled 'Synthesis Source Files Read' with the following data:

	File Name with User-Entered Path	Used in Netlist	File Type	File Name with
1	pipemult.vhd	yes	User VHDL File	C:/altera_trn/Quartus_Prime_Foundation_17_1..
2	mult.ip	yes	User File	C:/altera_trn/Quartus_Prime_Foundation_17_1..
3	mult/lpm_mult_170/synth/mult_lpm_mult_170_h3kjdy.v	yes	User Verilog HDL File	C:/altera_trn/Quartus_Prime_Foundation_17_1..ul
4	mult/synth/mult.vhd	yes	User VHDL File	C:/altera_trn/Quartus_Prime_Foundation_17_1..
5	ram.hex	yes	User Hexadecimal (Intel-Format) File	C:/altera_trn/Quartus_Prime_Foundation_17_1..
6	ram.ip	yes	User File	C:/altera_trn/Quartus_Prime_Foundation_17_1..
7	ram/ram_2port_171/synth/ram_ram_2port_171_nvmxoya.v	yes	User Verilog HDL File	C:/altera_trn/Quartus_Prime_Foundation_17_1..oi
8	ram/synth/ram.vhd	yes	User VHDL File	C:/altera_trn/Quartus_Prime_Foundation_17_1..
9	pipemult_lc.smf	yes	User Unspecified File	C:/altera_trn/Quartus_Prime_Foundation_17_1..
10	sm1.smf	yes	User Unspecified File	C:/altera_trn/Quartus_Prime_Foundation_17_1..
11	hex1.hex	yes	User Hexadecimal (Intel-Format) File	C:/altera_trn/Quartus_Prime_Foundation_17_1..

Example: Resource Usage

- Synthesis resource usage: estimates of FPGA resources required to implement design
- Fitter resource usage: detailed information on all resources used by design for each stage

The screenshot shows the 'Compilation Report' window for a project named 'Quartus_Prime_Foundation_17_1_v1/QPF17_1/VHDL/pipemult - pipemult_lc'. The 'Table of Contents' on the left includes sections like Parallel Compilation, Flow Summary, and Synthesis. The 'Synthesis Resource Usage Summary for Partition "root_partition"' table on the right provides a breakdown of resources:

Resource	Usage
Estimate of Logic utilization (ALMs needed)	121
Combinational ALUT usage for logic	219
-- 7 input functions	0
-- 6 input functions	0
-- 5 input functions	0
-- 4 input functions	120
-- <3 input functions	99
Dedicated logic registers	96
I/O pins	78
Total MLAB memory bits	0
Total block memory bits	2048
Total DSP Blocks	0
-- Total Fixed Point DSP Blocks	0
-- Total Floating Point DSP Blocks	0
Maximum fan-out node	clk1
Maximum fan-out	128
Total fan-out	1602

The screenshot shows the 'Fitter Resource Usage Summary' window. The 'Table of Contents' on the left lists various stages and their details. A red box highlights the 'Plan Stage' section, which includes 'Device Options', 'Operating Settings and Conditions', and 'Pin-Out File'. The main table on the right shows detailed resource usage for different categories:

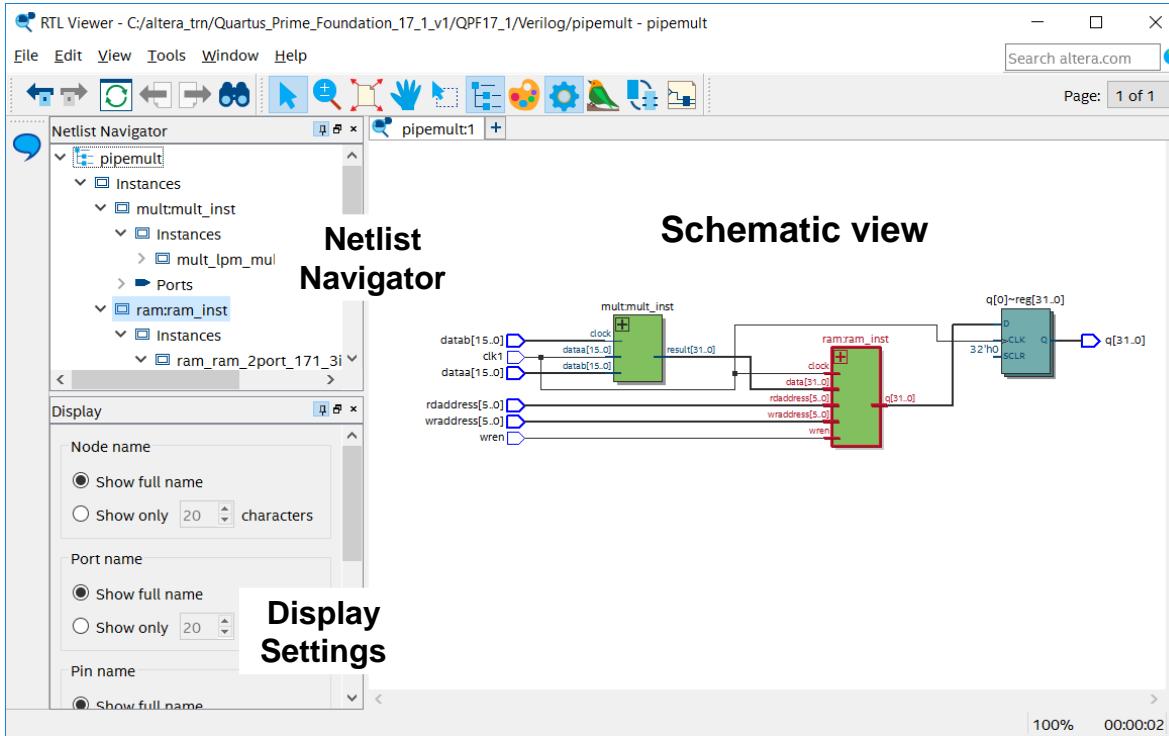
Resource	Usage	%
Logic utilization (ALMs needed / total ALMs on device)	116 / 427,200	< 1 %
ALMs needed [=A=B+C]	116	
[A] ALMs used in final placement [=a+b+c+d]	110 / 427,200	< 1 %
[a] ALMs used for LUT logic and registers	16	
[b] ALMs used for LUT logic	94	
[c] ALMs used for registers	0	
[d] ALMs used for memory (up to half of total ALMs)	0	
[B] Estimate of ALMs recoverable by dense packing	1 / 427,200	< 1 %
[C] Estimate of ALMs unavailable [=a+b+c+d]	7 / 427,200	< 1 %
[a] Due to location constrained logic	0	
[b] Due to LAB-wide signal conflicts	0	
[c] Due to LAB input limits	7	
[d] Due to virtual I/Os	0	
Difficulty packing design	Low	
Total LABs: partially or completely used	14 / 42,720	< 1 %
-- Logic LABs	14	
-- Memory LABs (up to half of total LABs)	0	
Combinational ALUT usage for logic	219	
-- 7 input functions	0	
-- 6 input functions	0	
-- 5 input functions	0	
-- 4 input functions	120	
-- <3 input functions	99	
Combinational ALUT usage for route-throughs	0	

Netlist Viewers

- RTL Viewer
 - Schematic of design after Analysis & Elaboration
 - Visually check initial HDL before synthesis optimizations
 - Locate synthesized nodes for assigning constraints
 - Debug verification issues
- Technology Map Viewers (Post-Mapping or Post-Fitting)
 - Graphically represents results of mapping (post-synthesis) & fitting
 - Analyze critical timing paths graphically
 - Locate nodes & node names after optimizations (cross-probing)

Netlist Viewers

Tools menu → Netlist Viewers

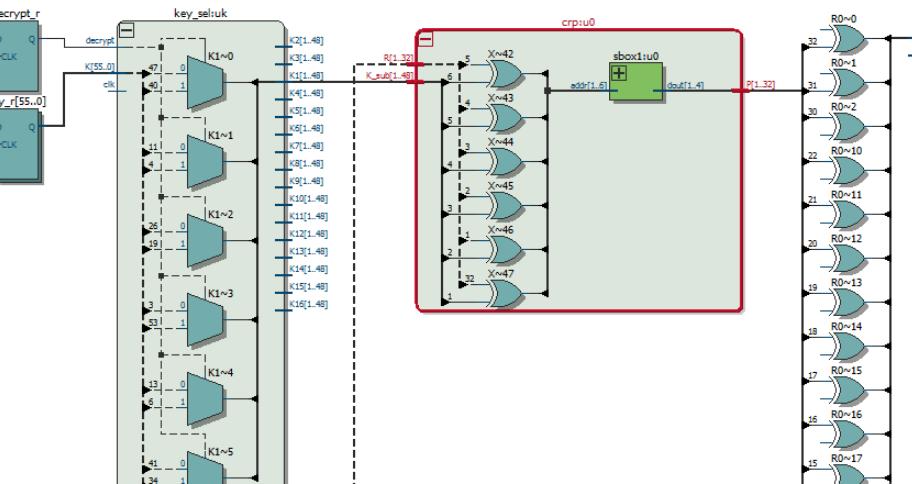


Note: Must perform elaboration first (e.g. Analysis & Elaboration OR Analysis & Synthesis)

Schematic View (RTL Viewer)

Represents design using logic blocks & nets

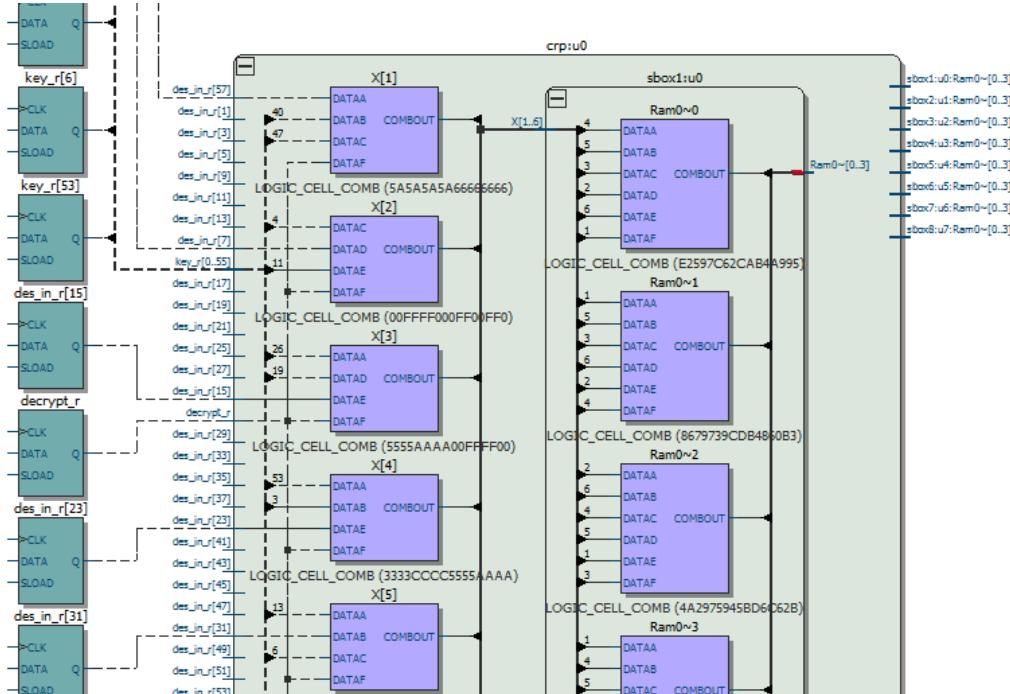
- I/O pins
- Registers
- Muxes
- Gates (AND, OR, etc.)
- Operators (adders, multipliers, etc.)



Schematic View (Technology Map Viewer)

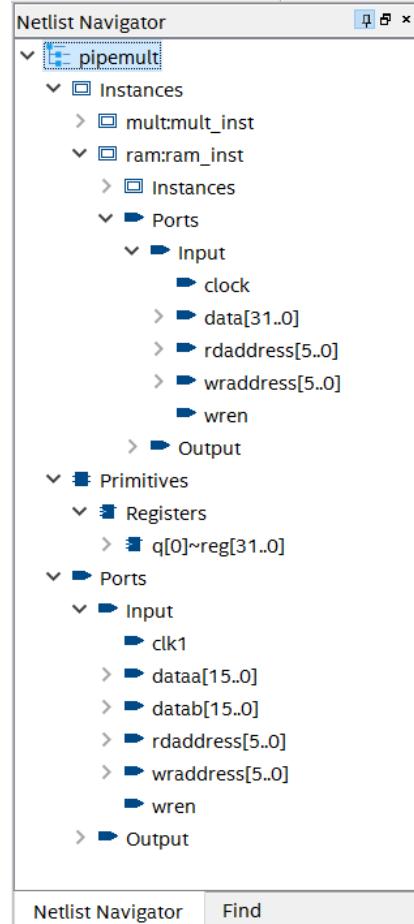
Represents design using atoms

- I/O pins & cells
- Logic cells (Lcells)
- Memory blocks
- MAC (DSP blocks)



Netlist Navigator (Hierarchy List)

- Traverse between levels of design hierarchy
- View logic schematic for each hierarchical level
- Break down each hierarchical level into netlist elements or atoms
 - Instances
 - Primitives
 - Registers
 - Buffers
 - Logic
 - Operators
 - Atoms
 - I/O
 - Ports
 - State machines



Using the Netlist Navigator

The screenshot shows the RTL Viewer interface. The title bar reads "RTL Viewer - C:/altera_trn/Quartus_Prime_Foundation_17_1_v1/QPF17_1/Verilog/pipemult - pipemult". The menu bar includes File, Edit, View, Tools, Window, and Help. The toolbar contains various icons for navigation and editing. The left panel is titled "Netlist Navigator" and shows a tree view of the design. Under "pipemult", there are three main categories: Instances, Primitives, and Ports. The "Instances" category is expanded, showing two entries: "multimult_inst" and "ram:ram_inst". A blue arrow points from the text "Expanding instances shows blocks within including:" to the "Instances" node in the tree. Another blue arrow points from the text "Highlighting object in Netlist Navigator highlights that element in schematic view" to the "ram:ram_inst" entry in the tree. The right panel displays the "Schematic" view. It shows a green rectangular block labeled "ram:ram_inst". On the left side of the block, there are five input ports: "clock", "data[31..0]", "rdaddress[5..0]", "wraddress[5..0]", and "wren". On the right side, there is one output port labeled "q[31..0]". A blue arrow points from the "ram:ram_inst" entry in the Netlist Navigator to the corresponding block in the Schematic view. The text "Highlighting object in Netlist Navigator highlights that element in schematic view" is displayed in bold black font above the schematic.

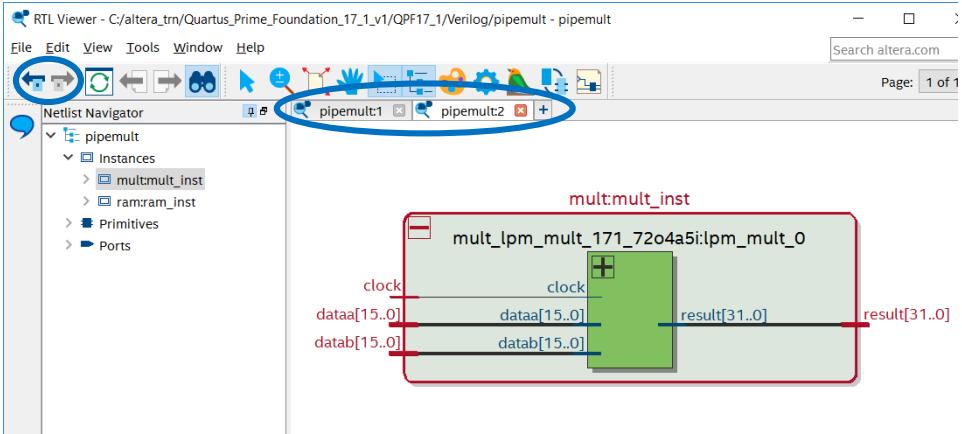
Expanding instances shows blocks within including:

- Instances
- Primitives
- Ports

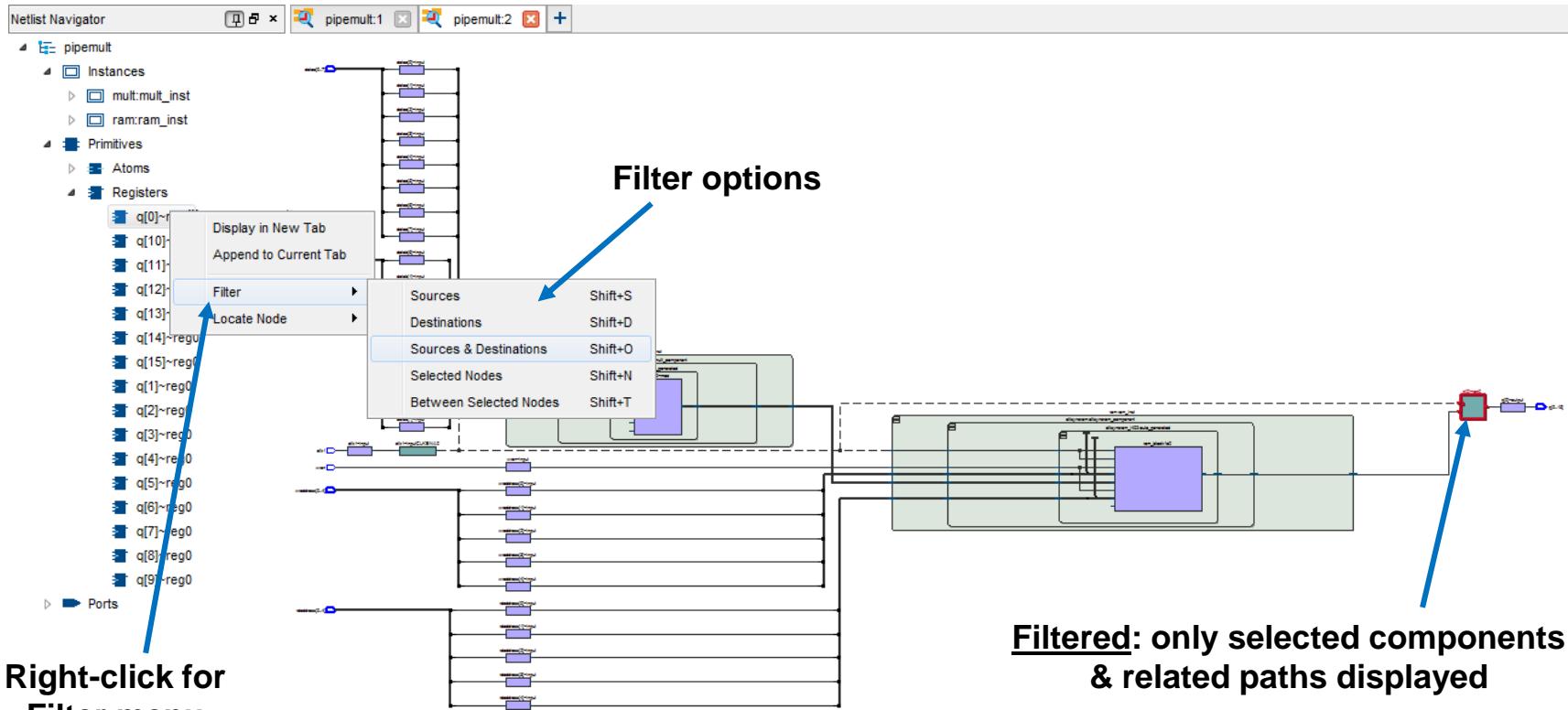
Highlighting object in Netlist Navigator highlights that element in schematic view

Using View Tabs

- Open multiple tabs to create and store particular views
- To create new tab:
 - Click +
 - Drag node(s) from Netlist Navigator to new empty tab or
 - Right-click on block in schematic or hierarchy and select **Display in New Tab**
- Tile or cascade tabs to see views simultaneously

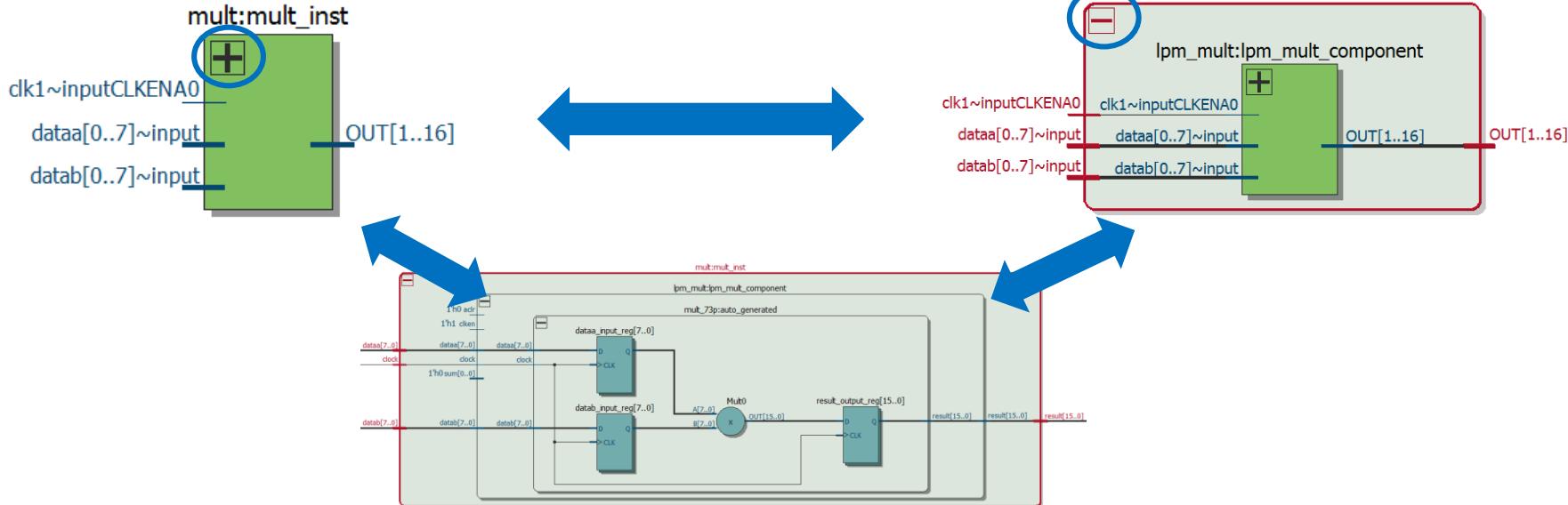


Filter Schematic



Instance Unwrap/Wrap

- Unwrap instance by clicking + to expand and see logic within
- Wrap instance by clicking - to hide expanded logic



Other Features (1)

- Bird's Eye View: overall panning view of design
- Block properties view (right-click)
 - Fan-in, fan-out, ports, parameters, logic diagram of atom
- Design search (Find in Edit menu or toolbar)
- Lookup table (LUT) internal detail as gate schematic or truth table
- Port/pin connectivity details: in-place view of all connections to/from port without filtering
- Shift-click instances in schematic, then drag and drop to reorganize view

Other Features (2)

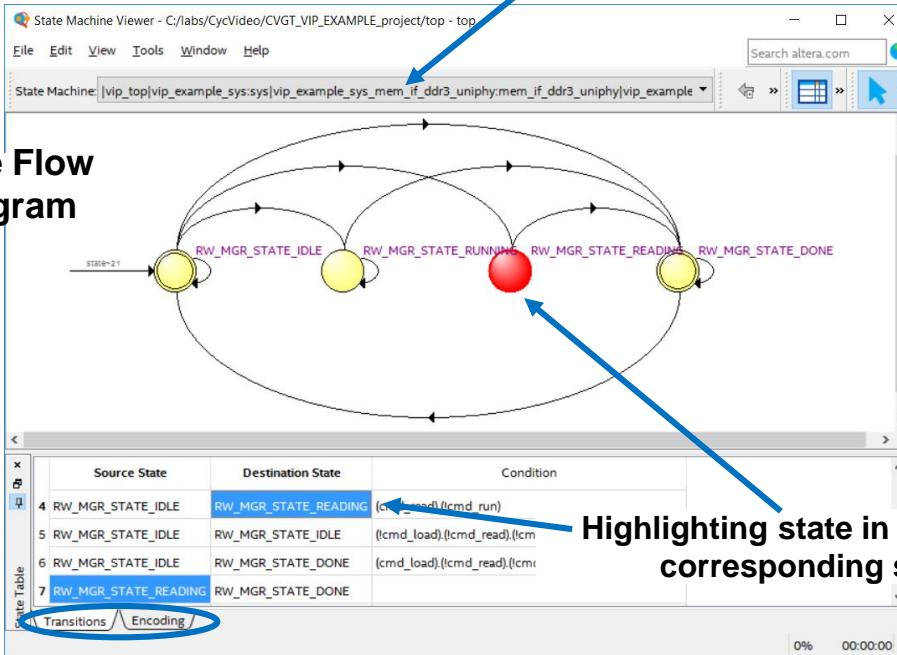
- Cross-probing: locate nodes from/to
 - Design files
 - Assignment Editor
 - Chip Planner
 - Resource Property Editor
 - RTL/Technology Map Viewers
 - Pin Planner
 - Timing Analyzer timing reports
 - Signal Tap logic analyzer

State Machine Viewer

- Tools menu → Netlist Viewers

Use drop-down to select from multiple state machines

State Flow
Diagram



Chip Planner

- Graphical view of design resource usage in target device
- Displays
 - Graphical layout of device resources
 - Routing channels between device resources
 - Global clock regions
- Uses
 - View placement of design logic
 - View connectivity between resources used in design
 - Make placement assignments
 - Debugging placement-related issues

Chip Planner



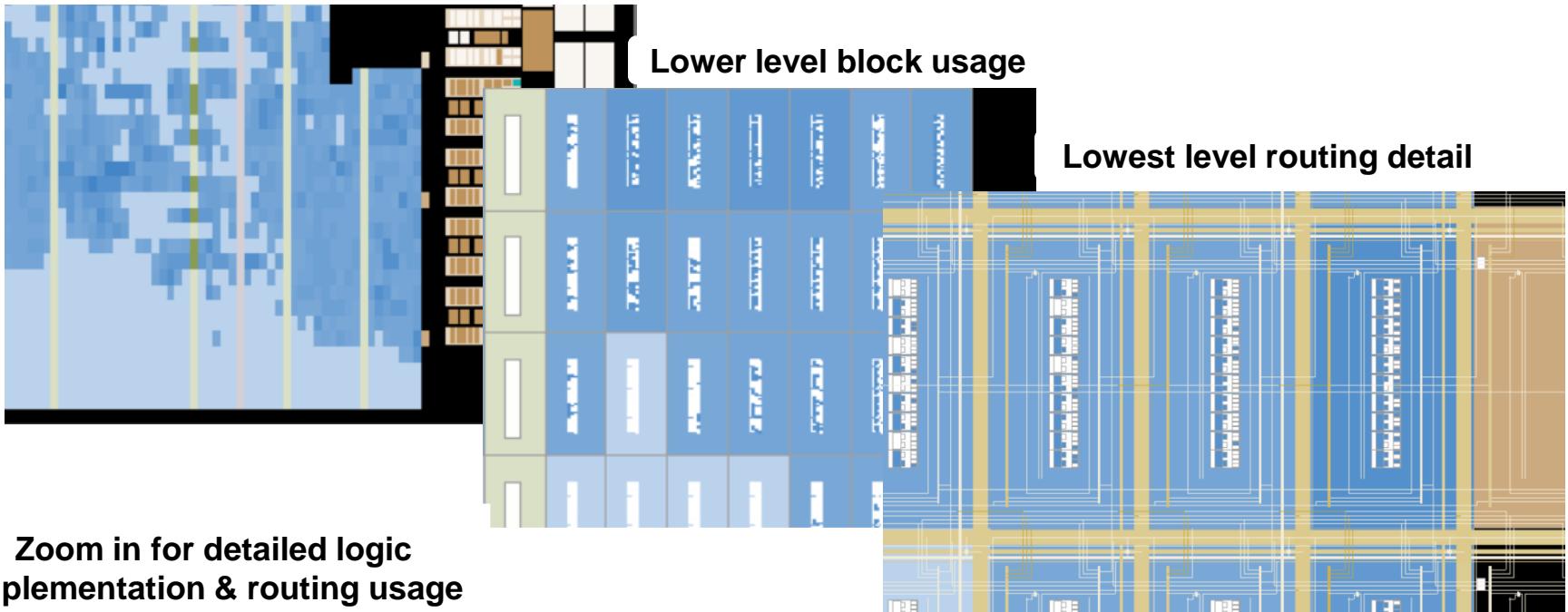
Tools menu or toolbar

The screenshot shows the Chip Planner application window with several labeled components:

- Report window**: A panel on the left containing a "Report not available" message and a toolbar with icons for report generation.
- Tasks window**: A panel on the left listing various tasks and reports, such as "Report Resources...", "Report Compilation Metrics", and "Mark Selection".
- Device floorplan aka Chip View**: The main workspace showing a grid of blue and green blocks. A blue rectangular area is highlighted and labeled "Memory block in use". An adjacent green area is labeled "Unused LAB".
- Editing Mode**: A dropdown menu set to "Assignment - 10AX115S2F45I1SG".
- Node Properties**: A panel on the right showing the properties of a selected element, "ram_block5a0". The "Properties/Modes" tab is active, displaying the full name as "dp_core_i|vip|vfb|vfb|pkt_trans_r". Other tabs include "Timing", "Located Objects", "Properties", "Fan-in", "Fan-out", and "Layers Settings".
- Selected Node Properties**: A sub-panel within the Node Properties window showing the detailed properties of the selected node.
- Layers Settings**: A tab in the Node Properties panel.

Floorplan Views

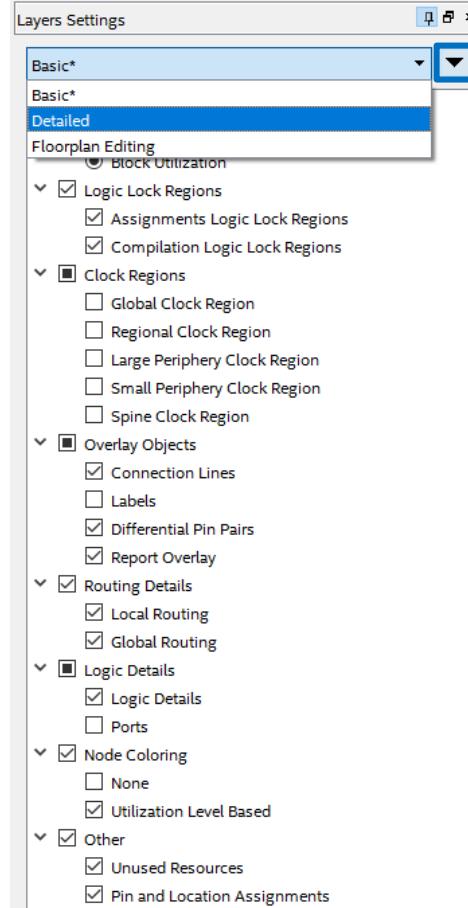
Overall device resource usage



Layers Settings

Quickly change which device resources are displayed in the background of the Chip View

- e.g. logic details, routing channels
- Default layers
 - Basic
 - Detailed
 - Floorplan Editing
- User-defined layers

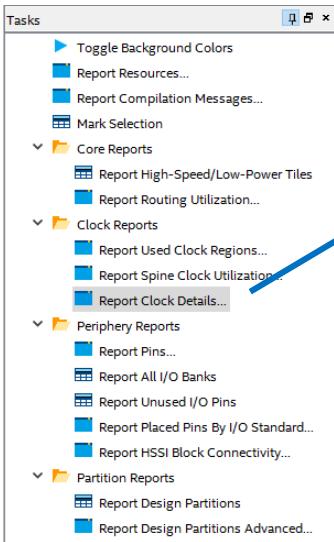


Report & Task Windows (1)

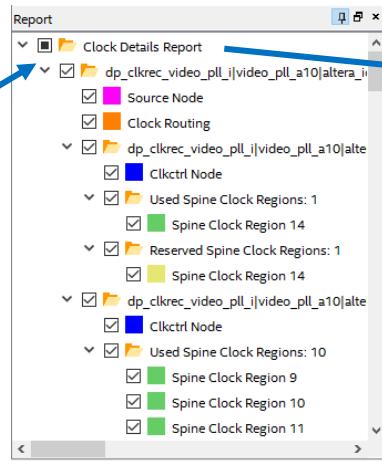
- Run tasks to generate reports that display information about device and device usage
 - e.g. resource location, clock region usage, transceiver connectivity
- Reports displayed as graphical overlays on floorplan

Report & Task Windows (2)

Double-click to run tasks that control information shown



Reports generated for each task run



Enable report to see overlay in Chip View

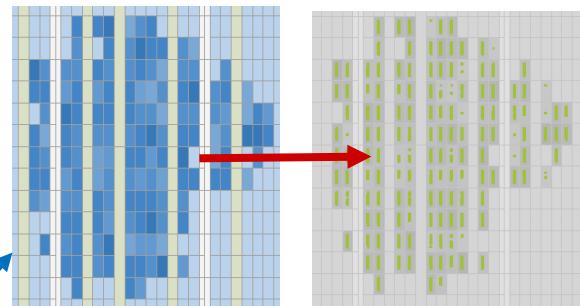
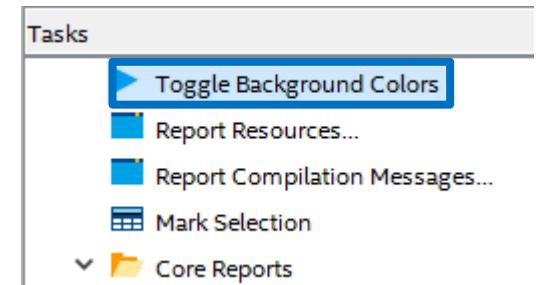
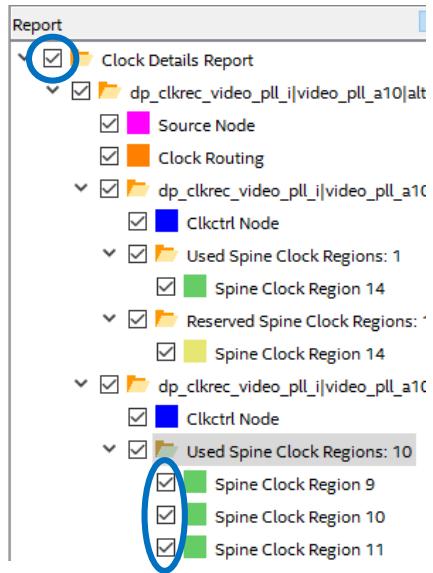
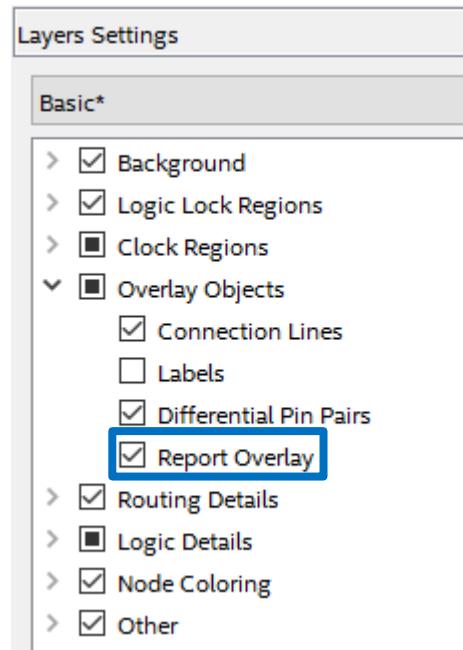


Controlling Report Overlay Visibility (1)

Quickly enable or disable report overlays as needed

- All overlays (**Layers Settings**)
- All overlays of a specific type (**Report** window)
- Individual overlays (**Report** window)

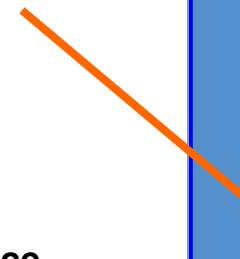
Controlling Report Overlay Visibility (2)



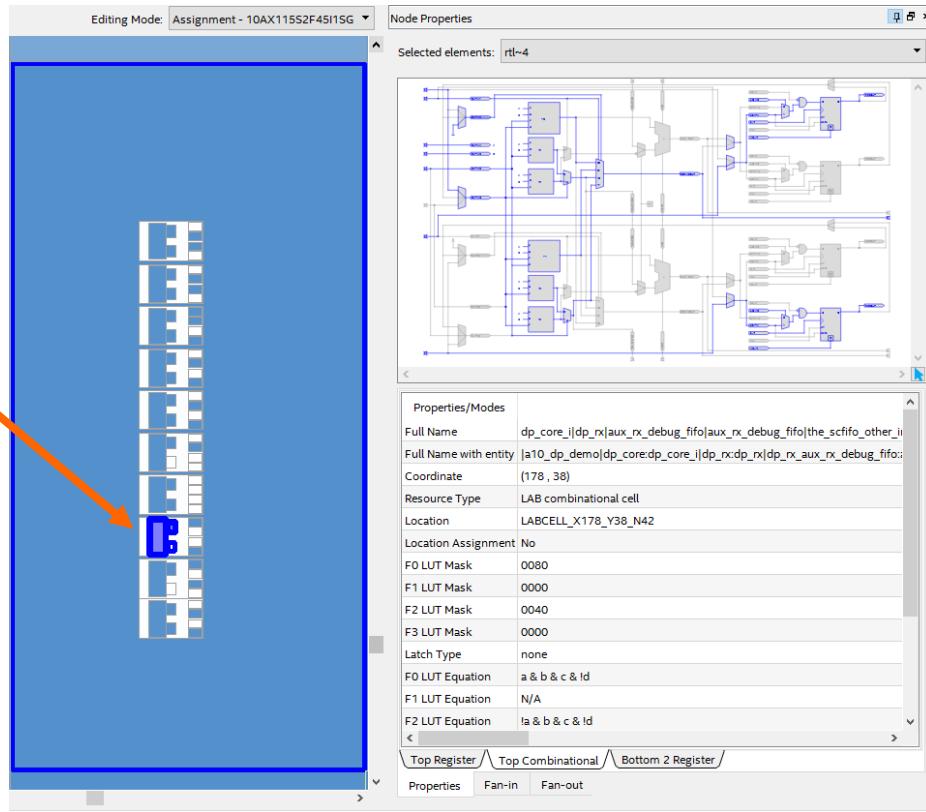
De-emphasize background colors to highlight report

Block Resource Properties

Click any block to view the internal resources detail



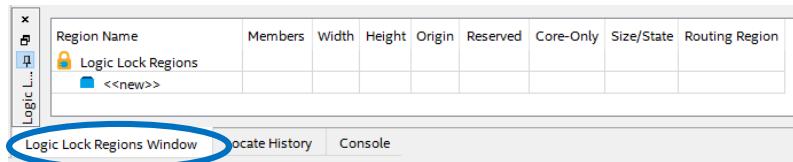
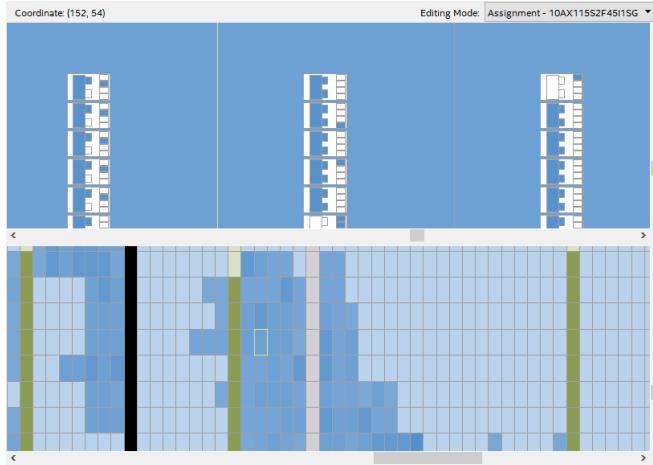
Double-clicking on the resource detail invokes the Resource Property Editor for making low-level changes



Splitter & Floorplanning Assignments

- Split the floorplan into two views
 - Allows for easier drag-and-drop operation
 - View at different zoom levels
- Ability to perform floorplanning related assignments directly in Chip Planner and see the effect in the floorplan
 - Location Assignments: manually place design elements into physical locations
 - Logic Lock Region: create physical partitions and assign logic for Fitter placement to each

Window menu → Split Window

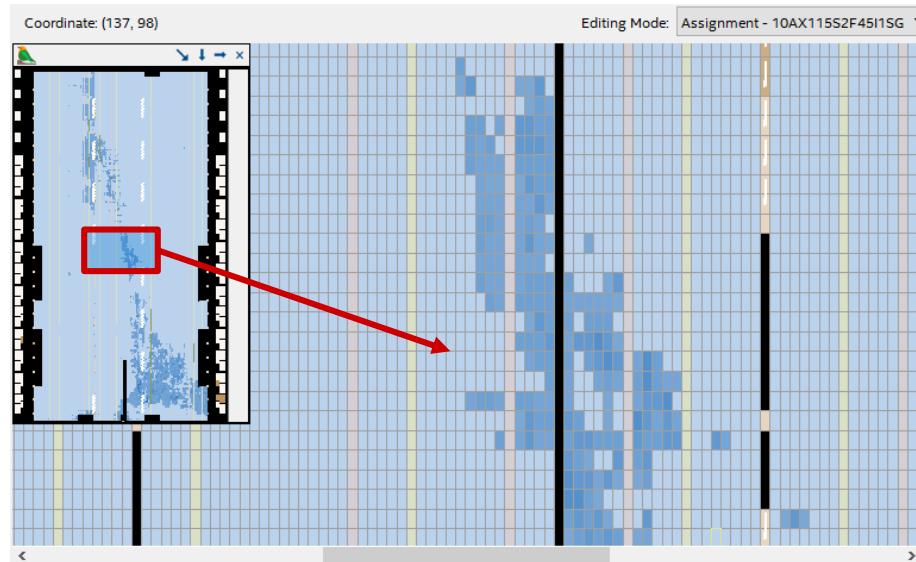


View menu → Location Assignments Window or Logic Lock Plus Region Window

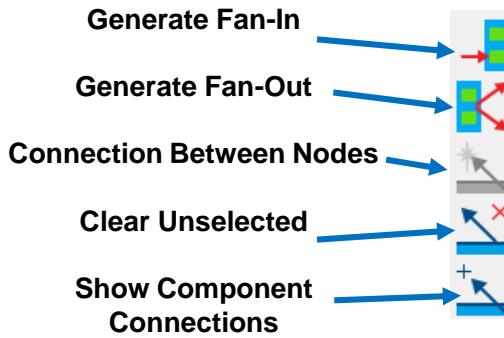
Bird's Eye View



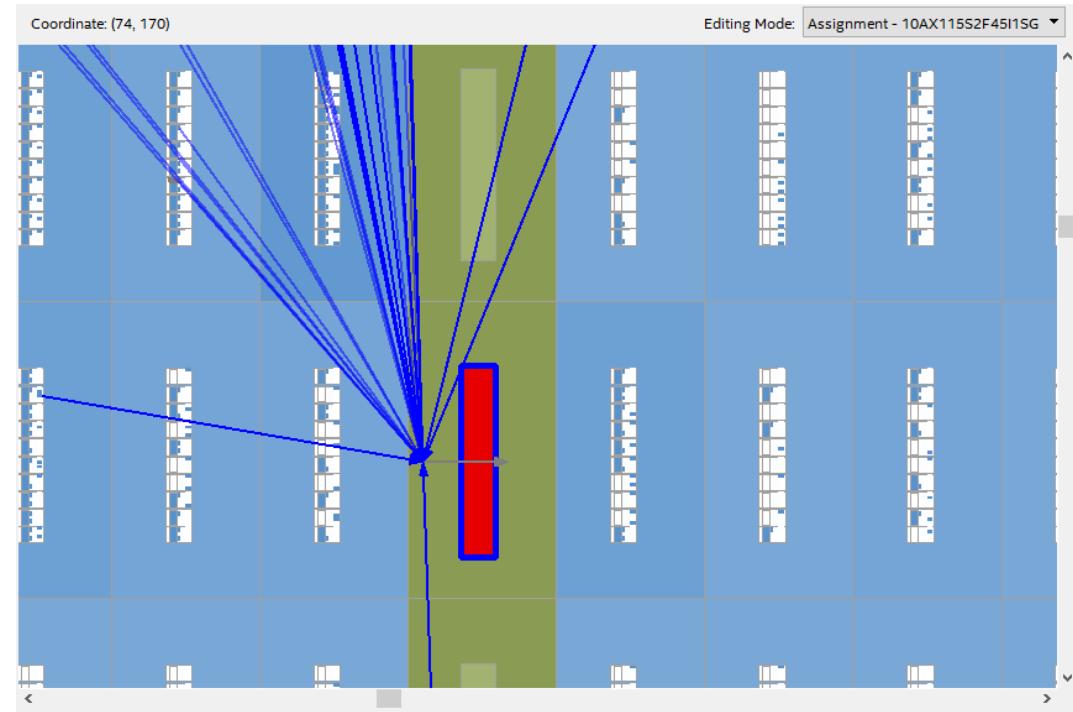
- Provides view of the entire device
- Navigate quickly through the floorplan



Displaying Fan-In & Fan-Out



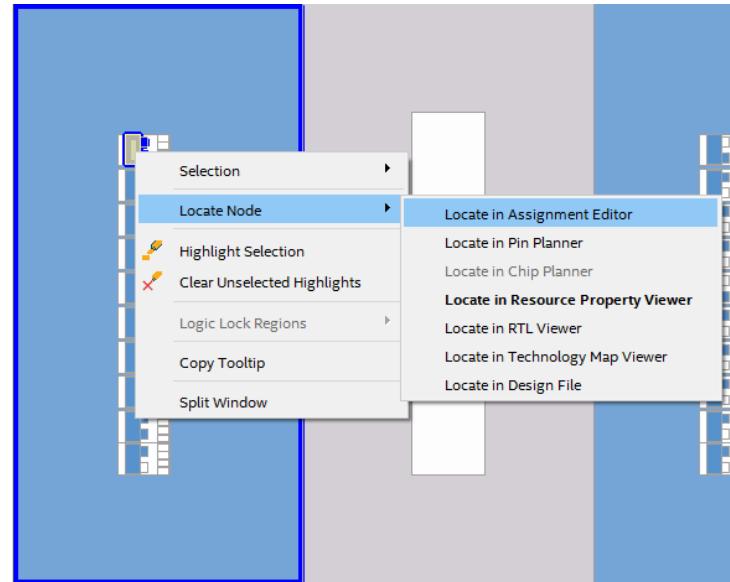
1. Highlight target blocks
2. Click fan-in toolbar button
3. Highlight target blocks again
4. Click fan-out toolbar button



Cross-Probing from/to Chip Planner

- Locate hierarchy blocks or specific logic from/to other Intel® Quartus® Prime Design Software tools
 - Project Navigator
 - Compilation Report
 - Design files
 - RTL Viewer
 - Technology Map Viewers
 - Messages window
 - Pin Planner
 - Timing Analyzer reports
 - Resource Property Editor

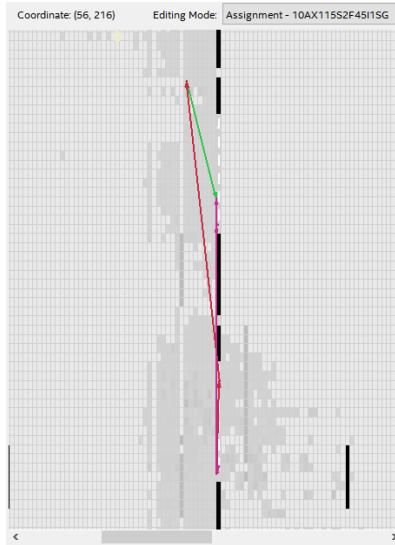
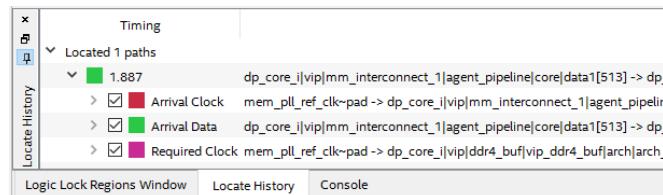
Right-click node & Locate in...



Cross-Probe Example

- Locate timing path from Timing Analyzer
 - Use placement to understand path timing
- Keep track of located paths using **Locate History** tab
- Quickly highlight (or re-highlight) entire or portion of timing path
- View actual signal routing path

Chip Planner View menu →
Locate History



Why Did I Receive Undesired Results?

- Incorrect coding
- Non-recommended coding style
- Suboptimal synthesis & fitting settings/constraints (*discussed next*)
- Use tools described to find problems and help fix them
- Note: For more details on optimizing designs based on undesired results, please attend the course [Timing Closure with the Intel® Quartus® Prime Software](#)

Exercise 3

Compilation Summary

- Compilation includes synthesis & fitting
- Compilation Report contains detailed information on compilation results
- Use Intel® Quartus® Prime Design Software tools to understand how design was processed
 - RTL Viewer
 - Technology Map Viewers
 - State Machine Viewer
 - Chip Planner

Compilation Support Resources

- Intel® Quartus® Prime Design Software Handbook chapters
 - Design Compilation (Volume 1)
 - Optimizing the Design Netlist (Volume 1)
 - Analyzing and Optimizing the Design Floorplan with the Chip Planner (Volume 2)
- Training courses
 - [Incremental Optimization with the Intel Quartus Prime Pro Edition Software](#) (online)
 - [Using the Intel Quartus Prime Software: Chip Planner](#) (online)
 - [Timing Closure with the Intel Quartus Prime Software](#) (instructor-led)



The Intel® Quartus® Prime Design Software: Foundation

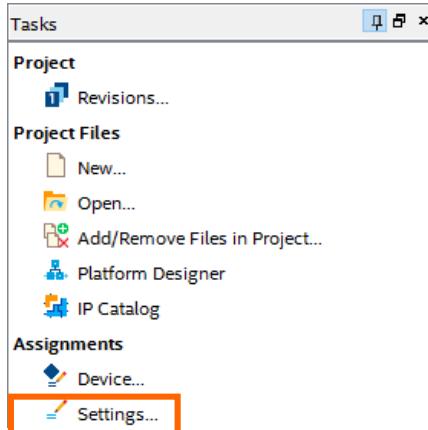
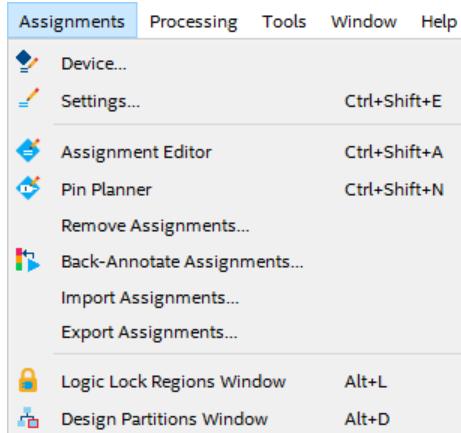
Settings & Assignments

Setting & Assignments Objectives

- Define the difference between settings and assignments
- List some examples of settings and assignments that can appear in an Intel® Quartus® Prime Design Software project
- Create settings and assignments using various methods in the Intel® Quartus® Prime software

Synthesis & Fitting Control

- Controlled using two methods
 - **Settings:** project-wide switches
 - **Assignments:** individual entity/node controls
- Both accessed in **Assignments** menu or **Tasks** window
- Stored in **.qsf** file for project/revision
- Timing constraints stored in separate **.sdc** file
 - Discussed in online and instructor-led Timing Analyzer classes



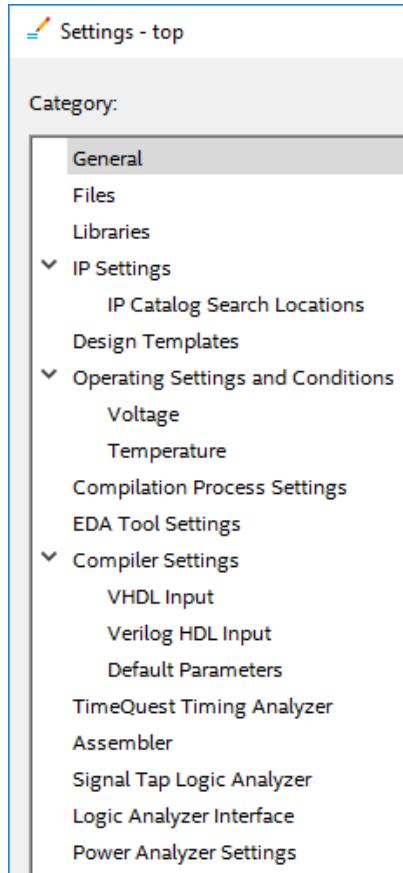
Settings

- Project-wide switches that affect entire design
- Examples
 - Device selection
 - Synthesis optimization
 - Fitter settings
 - Physical synthesis
- Located in **Device** and **Settings** dialog boxes
 - **Assignments** menu
 - **Tasks** window

Settings Dialog Box

Change settings

- Top-level entity
- Add/remove files
- Libraries
- Compiler settings
- EDA tool settings
- Fitter settings
- Timing Analyzer settings
- Power analysis settings

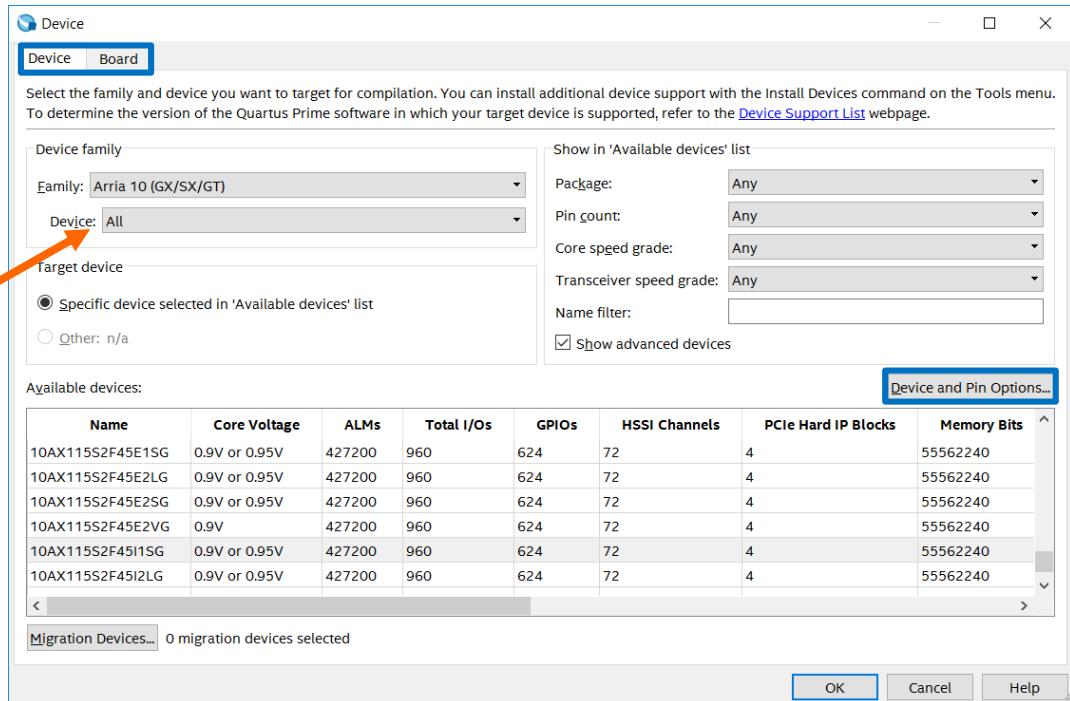


Tcl: `set_global_assignment -name <assignment_name> <value>`

Device Settings

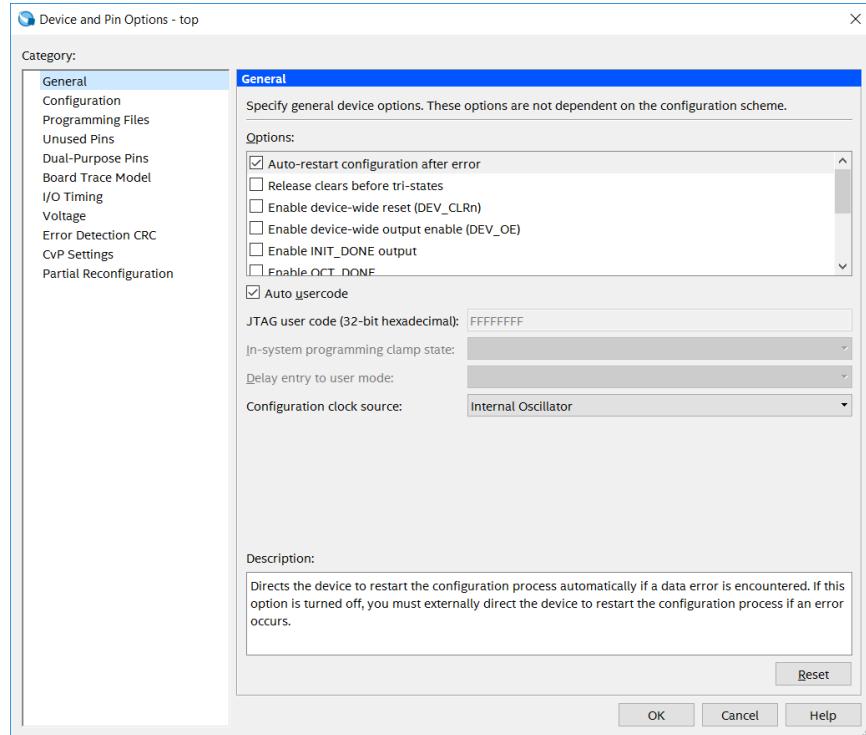
Device family must be installed along with Intel® Quartus® Prime Design Software

Assignments menu → Device or Tasks window



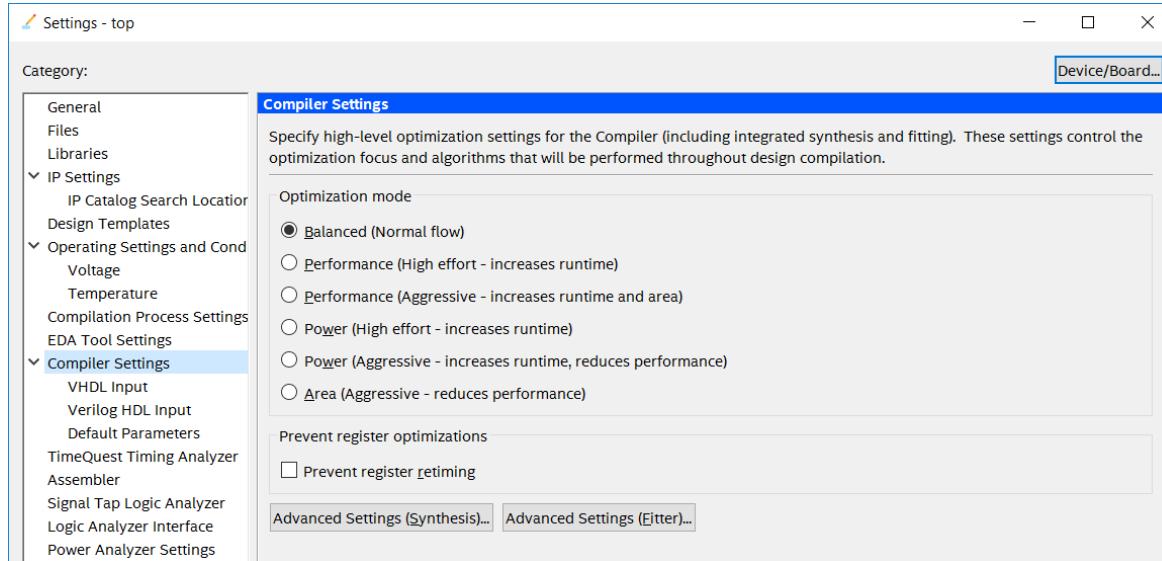
Device and Pin Options

- Hardware-specific feature settings



Settings Example: Consolidated Compiler Optimization Settings

- Choose general options for how compiler should optimize design
- Set specific optimization features under **Advanced Settings** buttons



Assignments (aka Logic Options, Constraints)

- Individual switches applied
 - I/O
 - Internal nodes
 - Hierarchical blocks (design entities)
- Assignment Editor manages assignments
- Must perform analysis & elaboration
- Example assignment: **Optimization Technique**



Assignment Editor

Provides spreadsheet assignment entry & display

- Copy & paste support
- Multi-cell editing

Sort on customizable columns

Assignments menu, toolbar, or Tasks window

Filter nodes

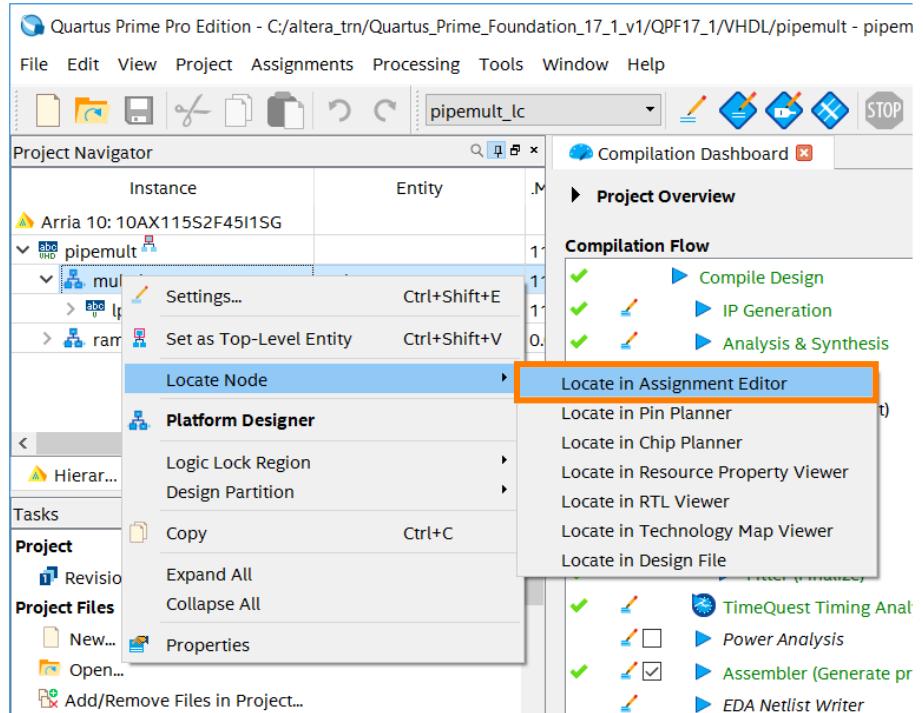
	tat	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	✓		in cpu...etn	Location	PIN_BD27	Yes			
2	✓		in refclk1_p	Location	PIN_AG37	Yes			
3	✓		in mem...clk	Location	PIN_F34	Yes			

This cell specifies the destination name for point-to-point assignments. For single-point assignments, this cell specifies the destination of the assignment. Intel recommends using the Node Finder to assign a destination name.

Enable/disable individual assignments

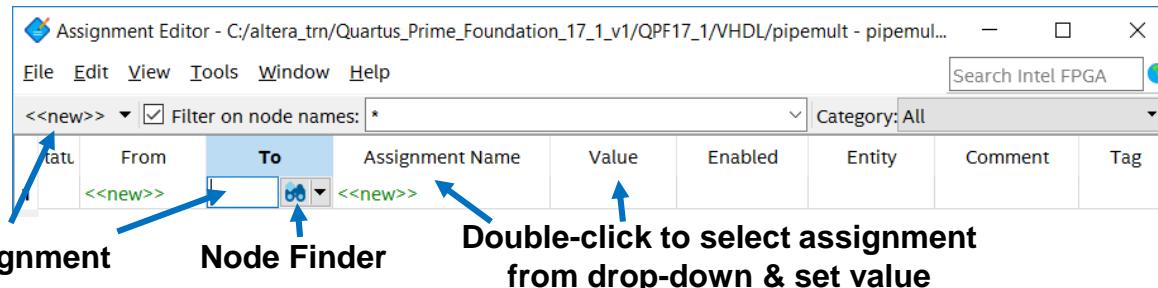
Creating Assignments: Cross-Probing

- Cross-probe (Locate Node) to Assignment Editor from:
 - Project Navigator
 - Message window
 - Compilation Report
 - Design files
- Assignment Editor Node Filter automatically filled in with cross-probed node(s)
- Easiest method for creating assignments



Creating Assignments: Assignment Editor

- Cross-probe from other tools to fill in filter
- Click on <<new>> button to create assignment for cross-probed node or entity
- Double-click <<new>> in the **From** or **To** columns to create a new assignment from scratch
 - Manually type in object name
 - Or click **Node Finder** icon to search





Node Finder

Search by name using wildcards (? or *)

Use filter to select the type of nodes to search/select

Locate nodes in a certain level of hierarchy

Show/hide filtering options

The screenshot shows the Node Finder dialog box with the following interface elements:

- Named:** A text input field containing a wildcard search term (*).
- Options**:
 - Filter:** A dropdown menu set to "Design Entry (all names)".
 - Look in:** A dropdown menu set to "pipemult".
 - Search**: An orange button.
 - Customize...**: A button.
 - Include subentities**: A checked checkbox.
 - Hierarchy view**: A checked checkbox.
- Matching Nodes:** A table listing nodes found in the hierarchy:

Name	Assignments
ram_inst	Unassigned
wren	Unassigned
mult_inst	Unassigned
clock	Unassigned
re.~0	Unassigned
re.10	Unassigned
- Nodes Found:** A table showing the selected node:

Name	Assignments
mult_inst clock	Unassigned
- Navigation:** A central area with arrows (up, down, left, right) for moving between matching nodes and found nodes.
- Message:** "Find completed successfully. Found 1407 nodes in 00 min 01 sec."
- Buttons:** "OK" and "Cancel".

Annotations with blue arrows and text callouts point to various features:

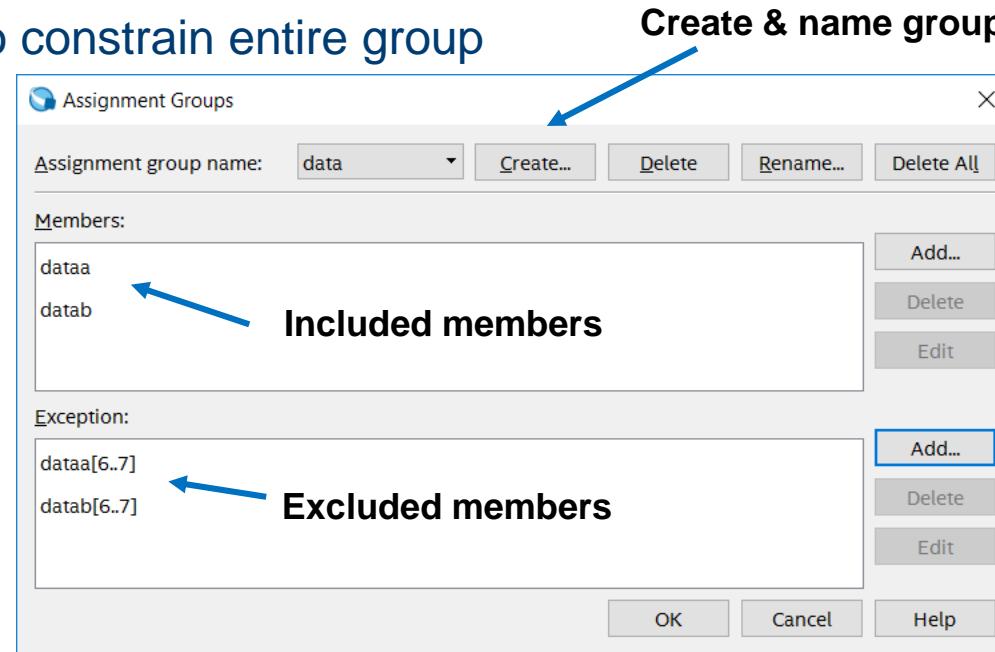
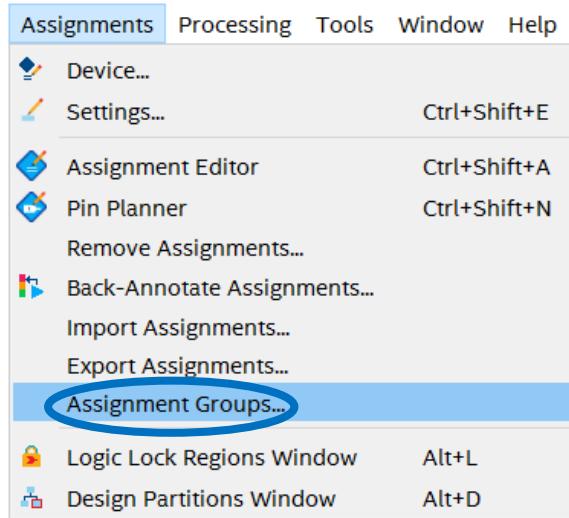
- An arrow points from the "Named" search field to the "Search" button.
- An arrow points from the "Filter" dropdown to the "Customize..." button.
- An arrow points from the "Hierarchy view" checkbox to the "Show/hide filtering options" text.
- An arrow points from the "Nodes Found" table to the "Tooltip with hierarchy path to target node" text.
- An arrow points from the "Matching Nodes" table to the "List of found nodes arranged by hierarchy" text.
- An arrow points from the navigation arrows in the center to the "Select node(s) on left & use arrows to move to the right" text.

List of found nodes arranged by hierarchy

Select node(s) on left & use arrows to move to the right

Assignment Groups

- Assign names to user-defined groups of nodes
- Allows single assignment to constrain entire group



Updating .qsf File

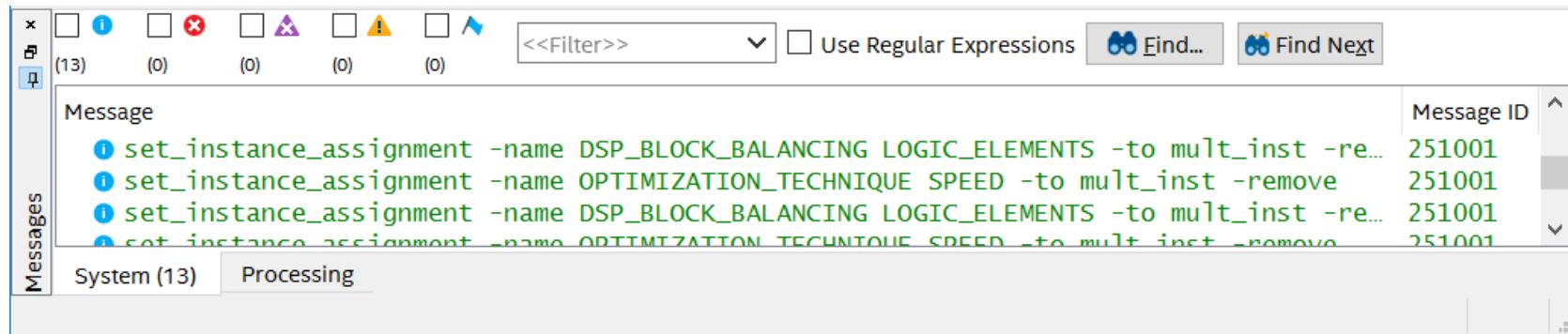
- By default, **.qsf** not updated automatically when constraint entered
- **.qsf** updated only when
 - Project is saved or closed (**File** menu)
 - Assignment Editor is saved or closed
 - Beginning of any processing task (e.g. compilation)
 - Any IP parameter editing tool is launched
 - Changing revisions
- Change behavior to update assignments immediately (**Tools** menu → **Options** → **General** → **Processing**)
 - May impact software performance slightly due to file accesses

Assignment Tcl Commands

Equivalent Tcl commands displayed as assignments are entered

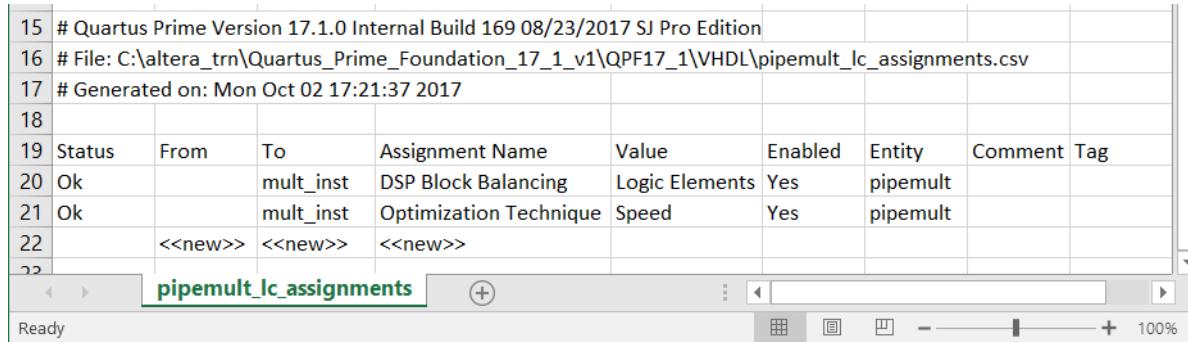
- Manually copy to create Tcl scripts
- Export command (**File** menu) writes all assignments to a Tcl file

Messages window



Export Assignments

- Export to **.csv** file (**File** menu → **Export**)
 - Import and edit assignments in Excel



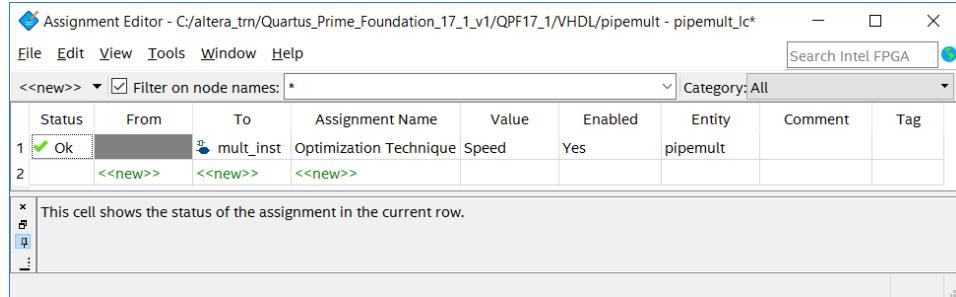
The screenshot shows the Quartus Prime software interface with a CSV file open. The file contains assignment data for a project named 'pipemult_lc_assignments'. The data includes header information and two rows of assignment details.

	Status	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
15	Ok		mult_inst	DSP Block Balancing	Logic Elements	Yes	pipemult		
16	Ok		mult_inst	Optimization Technique	Speed	Yes	pipemult		
17	<<new>>	<<new>>	<<new>>						
18									
19									
20									
21									
22									
23									

- Export to new **.qsf** (**Assignments** menu → **Export Assignments**)
 - Migrate assignments to another project

Assignment Example: Optimization Technique

- Selects synthesis optimization goal
(Speed, Balanced, Area)
- Apply to hierarchical entities
 - Locate (cross-probe) from Project Navigator
 - Or drag and drop into Assignment Editor
- May also apply project-wide in **Advanced Synthesis** Settings
- Affects synthesis & logic mapping; Intel® Quartus® Prime software synthesis only



Available Logic Options (Assignments)

The screenshot shows a web browser displaying the Quartus Prime Pro Edition Help version 17.1. The title bar reads "Quartus Prime Pro Edition Help version 17.1". The left sidebar has a "Content" tab selected, which is highlighted with an orange box. A blue arrow points from the text "Select ‘Logic Options Definition’ in Intel® Quartus® Prime Design Software Help Contents" to this "Content" tab. Below it, another orange box highlights the "Logic Options" section under the "Content" tab. The main content area is titled "Logic Options" and contains the text: "The following logic options are available in Intel® Quartus® Prime Pro Edition software." It lists several categories: Advanced logic options, Global Signals logic options, Synchronize selections between tools, I/O Timing logic options, Synthesis logic options, Simulation logic options, Fitter Optimization, Others, and All Logic Options. The "All Logic Options" link is also highlighted with an orange box. At the bottom of the page, there are links to Contact Intel, Terms of Use, Trademarks, Privacy, and Send Feedback.

Select “Logic Options Definition” in Intel® Quartus® Prime Design Software Help Contents

Logic Options

The following logic options are available in Intel® Quartus® Prime Pro Edition software.

- [Advanced logic options](#)
- [Global Signals logic options](#)
- [Synchronize selections between tools](#)
- [I/O Timing logic options](#)
- [Synthesis logic options](#)
- [Simulation logic options](#)
- [Fitter Optimization](#)
- [Others](#)
- [All Logic Options](#)

Contact Intel | Terms of Use | Trademarks | Privacy | Send Feedback

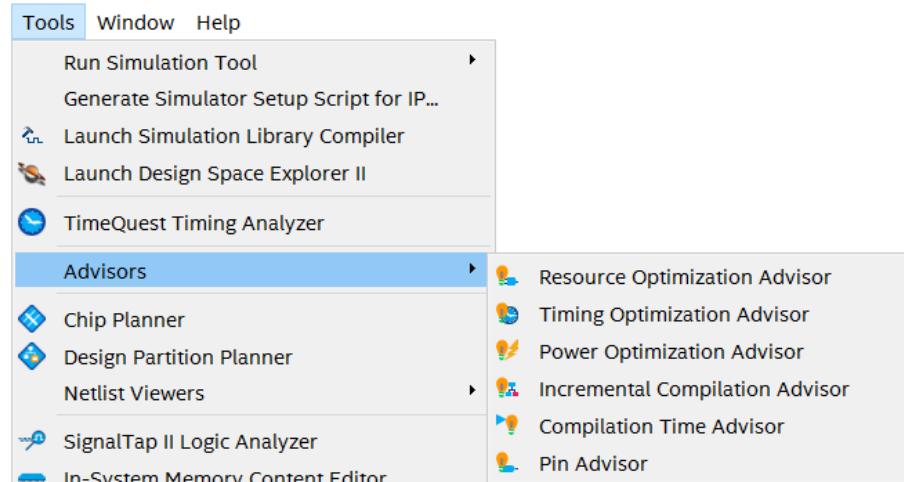
Copyright© Intel Corporation. All rights reserved. INTEL, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS, STRATIX, and all other brands, unless noted otherwise, and/or trademarks of Intel Corporation in the U.S. and other countries.

Links to all available assignments
organized by category with lists
of supported devices

Advisors

Provide project-specific recommendations (feedback) on optimizing designs

- Access through **Tools** menu
- Six types
 - Resource Optimization
 - Timing Optimization
 - Power Optimization
 - Incremental Compilation
 - Compilation Time
 - Pin
- Recommendations can contradict each other between different advisors



Advisor Example

The screenshot shows the Timing Optimization Advisor interface. On the left, a tree view lists various optimization categories. A green checkmark next to 'Maximum Frequency (fmax)' indicates it's being followed. A blue arrow points from the 'Links to locations in software to change settings/assignments manually' text to the 'Assignment Name' column in the table on the right. Another blue arrow points from the 'Current Setting' column to the 'Current Global Settings' text below it.

Timing Optimization Advisor - C:/altera_trn/Quartus_Prime_Foundation_17_1_v1/QPF17_1/VHDL/pipemult - pipe...

File Edit Tools Window Help

Timing Summary

- How to use the Timing Optimization...
- General Recommendations
 - Get more information
 - Create a revision
 - Review timing constraints - TQ
 - Select a Faster Speed Grade De...
- Maximum Frequency (fmax)
 - Use High-Effort fmax Optimizati...
 - Optimize for speed
 - Use physical synthesis optimi...
 - Enable Logic Cell Insertion - ...
 - Set maximum router effort
 - Avoid unrelated logic registe...
 - Other Recommendations
- I/O Timing (tsu, tco, tpd)
 - Use timing-driven compilation
 - Turn on Auto Global Clock
 - Use clock options
 - Use fast input, fast output, and ...

Summary

The following areas will be affected by the recommended changes:

- + Delay may decrease (fmax may increase)
- Logic element usage may increase
- = Compilation time is unaffected

Action

For Quartus Prime Integrated Synthesis, choose Speed under Optin Advanced Analysis & Synthesis Settings page under Compiler Setti box (Assignments menu). It is also recommended to set the optimi: Balanced if it is currently set to Area. Balanced technique gives bett: than Speed. But resource usage is better than with Speed (worse thi: specify the Optimization Technique logic option for individual partitions in your design using the Assignment Editor (Assignments menu), while leaving the project Optimization Technique setting at Balanced (for the best trade off between area and speed for certain device families) or Area (if area is an important concern).

Show 10 entries

Search:

Node Name	Assignment Name	Current Setting
mult_inst	Optimization Technique	SPEED

Showing 1 to 1 of 1 entries

Previous 1 Next

Current Global Settings:
Optimization Technique = BALANCED (Recommended: SPEED)

[Open Settings dialog box - Advanced Analysis & Synthesis Settings dialog box](#)
[Open Assignment Editor](#)

Green checkmark
indicates recommendation
already being followed

Links to locations in software to
change settings/assignments manually

Part of design is following
recommendation

Exercise 4

Settings & Assignments Summary

- Settings & assignments allow a designer to control how a design is synthesized and placed & routed
- Use the Settings dialog box to adjust project-wide settings
- Use the Assignment Editor to enable/disable individual assignments targeting hierarchical blocks, internal nodes, or I/O
- Optimization Advisors help improve design results through setting and assignment recommendations

Design Analysis Support Resources (1)

Intel® Quartus® Prime Design Software Handbook chapters (all Volume 2)

- Constraining Designs
- Design Optimization Overview
- Reducing Compilation Time
- Timing Closure & Optimization
- Power Optimization
- Area Optimization
- Netlist Optimizations & Physical Synthesis

Design Analysis Support Resources (2)

- Training courses
 - [The Intel® Quartus® Prime Design Software Series: Timing Analysis](#) (instructor-led)
 - [Advanced Timing Analysis with Timing Analyzer](#) (instructor-led)
 - [Timing Closure with the Intel Quartus Prime Software](#) (instructor-led)
 - [Timing Analyzer](#)
 - [Using Design Space Explorer](#)
 - [Timing Closure Using Intel Quartus Prime Software Physical Synthesis Optimizations](#)
 - [Resource Optimization](#)
 - [Power Optimization and Analysis](#)



The Intel® Quartus® Prime Design Software: Foundation

I/O Planning

I/O Planning Need

- I/O standards increasing in complexity
- FPGA/CPLD I/O structure increasing in complexity
 - Results in increased pin placement guidelines
- PCB development performed simultaneously with FPGA design
- Pin assignments need to be verified earlier in design cycle

Creating I/O-Related Assignments

- Pin Planner
- Interface Planner
 - For Intel® Arria® 10 and newer devices
- Import from spreadsheet in **.csv** format
- Type directly into **.qsf** file
- Directly in HDL code
- Scripting



The Intel® Quartus® Prime Design Software: Foundation

I/O Planning with the Pin Planner

Pin Planner

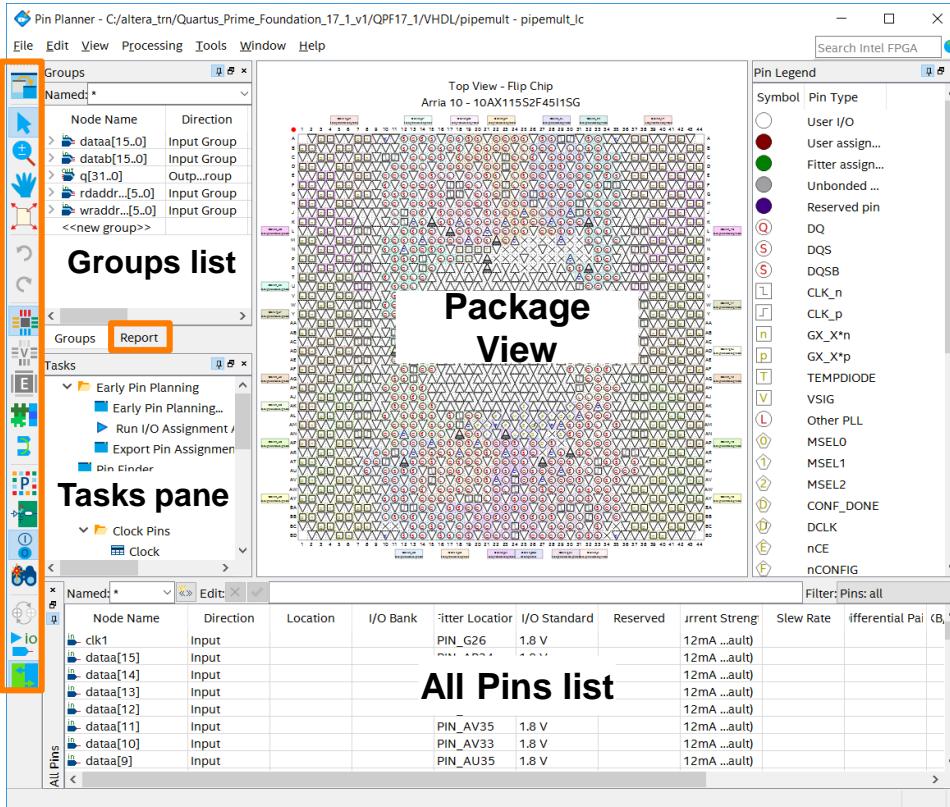


- Interactive graphical tool for assigning pins
 - Drag & drop pin assignments
 - Set pin I/O standards
 - Reserve future I/O locations
- Default window panes
 - Package View
 - All Pins list
 - Groups list
 - Tasks window
 - Report window

Assignments menu → **Pin Planner**, toolbar, or **Tasks** window

Pin Planner Window

Toolbar



Pin Planner Window Panes (1)

- Package View

- Displays graphical representation of chip package
 - Locate, make, or edit I/O assignments

- All Pins list

- Displays I/O pins (signals) in design
 - Edit pin assignments

Pin Planner Window Panes (2)

- Groups list
 - Similar to All Pins list displaying only groups & buses
 - Make bus and group assignments
 - Create new user-defined groups
- Tasks pane
 - Perform tasks such as Early Pin Planning and pin highlight reports
- Report Pane
 - Quickly enable/disable reports generated in the Tasks pane

Assigning Pin Locations Using Pin Planner

The screenshot shows the Pin Planner software interface. At the top, there's a grid of pins with various symbols like S, Q, V, T, L, etc., each with a color-coded background (green, orange, red, blue). A specific pin in the center-left is highlighted with a blue selection box. Below the grid is a table titled "All Pins".

Node Name	Direction	Location	I/O E
clk1	Input	PIN_R14	3E
dataa[15]	Input		
dataa[14]	Input		
dataa[13]	Input		
dataa[12]	Input		
dataa[11]	Input		
dataa[10]	Input		
dataa[9]	Input		

An orange arrow points from the highlighted pin in the grid to the first row of the table, indicating the action of dragging and dropping a single signal.

Drag & drop single top-level I/O signal/pin

Choose one-by-one or pin alignment direction (Edit menu)

This screenshot shows the same Pin Planner interface after multiple pins have been selected. A large cluster of pins in the center-right area is highlighted with a blue selection box. Below this, another table titled "All Pins" shows the results of the pin assignments.

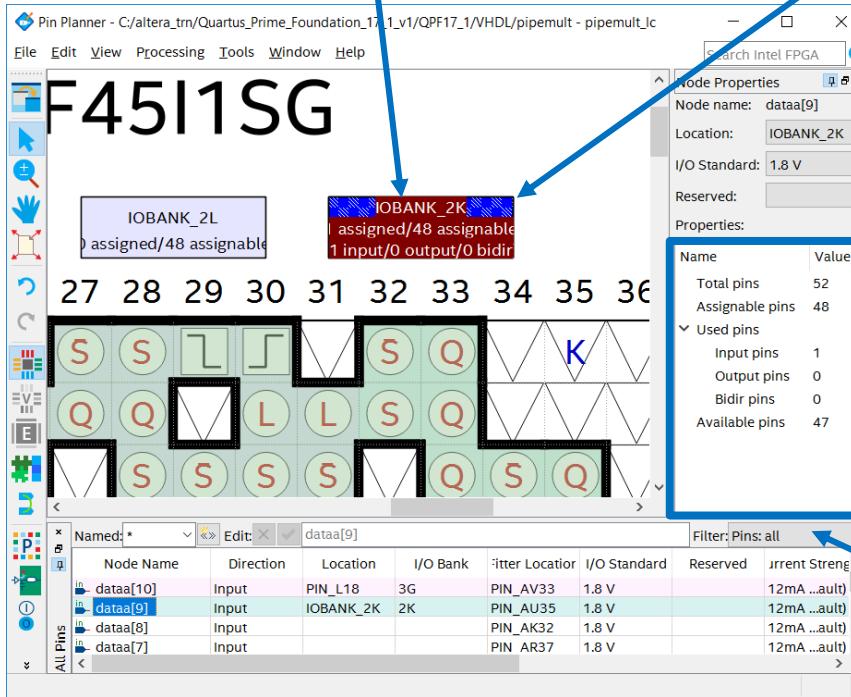
Node Name	Direction	Location	I/O E	Pin
clk1	Input	PIN_R14	3E	PIN_G2
dataa[15]	Input	PIN_M16	3G	PIN_AF
dataa[14]	Input	PIN_M15	3G	PIN_AL
dataa[13]	Input	PIN_L15	3G	PIN_AY
dataa[12]	Input	PIN_M17	3G	PIN_AL
dataa[11]	Input	PIN_M18	3G	PIN_AV
dataa[10]	Input	PIN_L18	3G	PIN_AV
dataa[9]	Input			PIN_AV

An orange arrow points from the highlighted pins in the grid to the second row of the table, indicating the action of dragging and dropping multiple signals.

Drag & drop multiple highlighted pins or buses

Assigning Pin Locations Using Pin Planner (2)

Drag & drop to I/O bank, VREF block, or device edge



Click pin or I/O bank to display pin properties

Filter nodes displayed in All Pins list

Assigning Pin Locations Using Pin Planner (3)

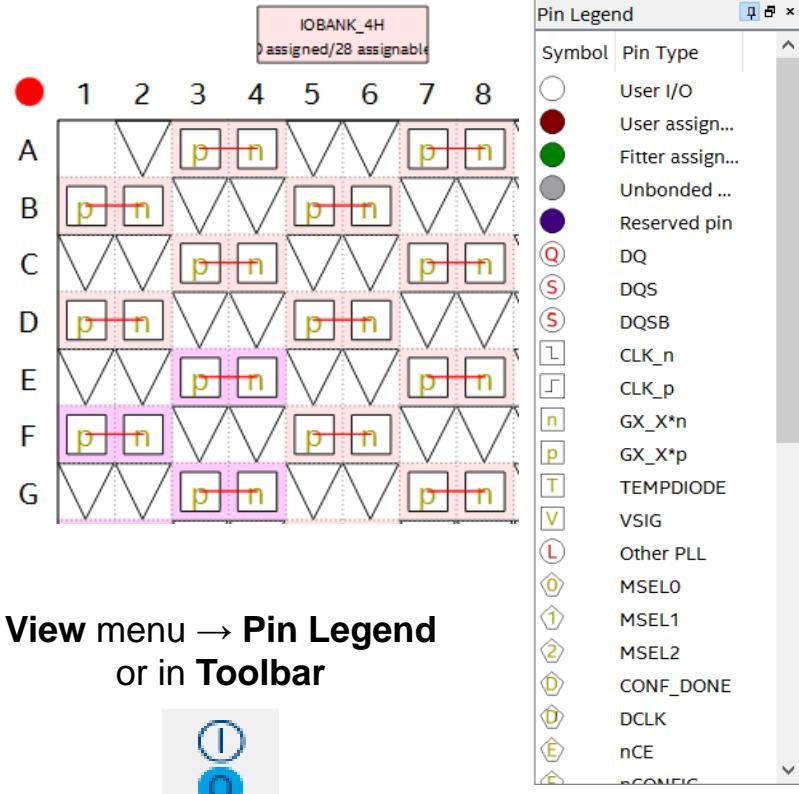
- Select available locations from list of pins color-coded by I/O bank

Node Name	Direction	Location	I/O Bank	Flitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	IOB/VCCT_G
in - dataaa[9]	Input	IOBANK_2K	2K	PIN_AU35	1.8 V		12mA ...ault)			
in - dataaa[8]	Input									
in - dataaa[7]	Input	PIN_A29	IOBANK_2L	Column I/O CLK_2L_0n, LVDS2L_13n, DQ2L						
in - dataaa[6]	Input	PIN_A30	IOBANK_2L	Column I/O CLK_2L_0p, LVDS2L_13p, DQ2L						
in - dataaa[5]	Input	PIN_A32	IOBANK_2K	Column I/O LVDS2K_2n, DQ4L						
in - dataaa[4]	Input	PIN_A33	IOBANK_2K	Column I/O LVDS2K_3n, DQ4L						
in - dataaa[3]	Input	PIN_AA13	IOBANK_3E	Column I/O PLL_3E_CLKOUT0p, PLL_3E_CLKOUT0, PLL_3E_FB0, LVDS3E_15p, DQ46R						
in - dataaa[2]	Input	PIN_AA14	IOBANK_3E	Column I/O LVDS3E_18n, DQ46R						
in - dataaa[1]	Input	PIN_AA30	IOBANK_2J	Column I/O LVDS2J_6n, DQ8L						
in - dataaa[0]	Input	PIN_AA32	IOBANK_2J	Column I/O LVDS2J_4n, DQSsn8L						
in - datab[15]	Input	PIN_AA33	IOBANK_2J	Column I/O LVDS2J_16n, DQSsn10L						
in - datab[14]	Input	PIN_AA34	IOBANK_2J	Column I/O LVDS2J_16p, DQS10L						

Pin Legend and Pin Planner Views

- Displays (View menu → **Show**, toolbar buttons, or right-click in Package View)
 - Device edges
 - I/O banks
 - VREF groups
 - Differential pin pairing
 - DQ/DQS and hard memory interface pins
 - PCIe* hard IP pins
- Easy-to-read pin legend

I/O banks & differential pairing

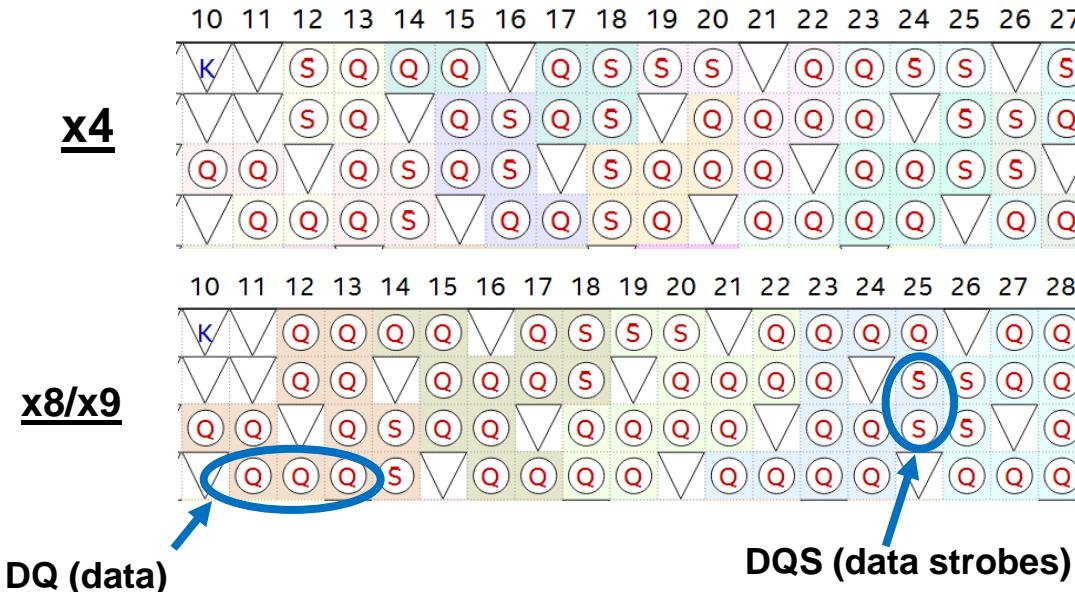


**View menu → Pin Legend
or in Toolbar**

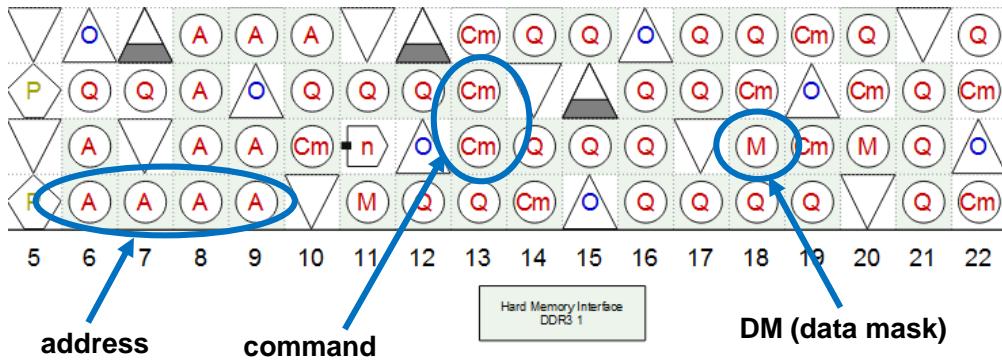


Example View: Show DQ/DQS Pins

- Show color-coded DQ/DQS sets in x4, x8/x9, x16/x18, or x32/x36 modes in the Package View for DDR interfaces



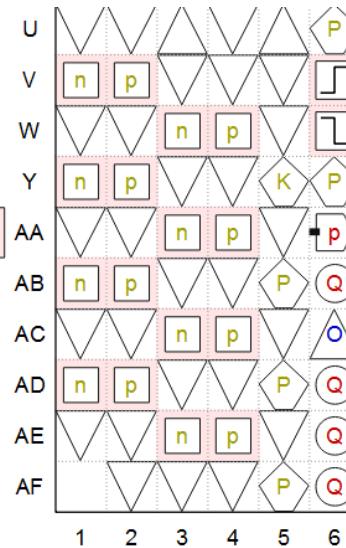
Example Views: Show Dedicated Hard Resources I/O



Hard external memory interface I/O

Hard PCIe* RX/TX/REFCLK I/O

PCIe Hard IP Interface x4 Let

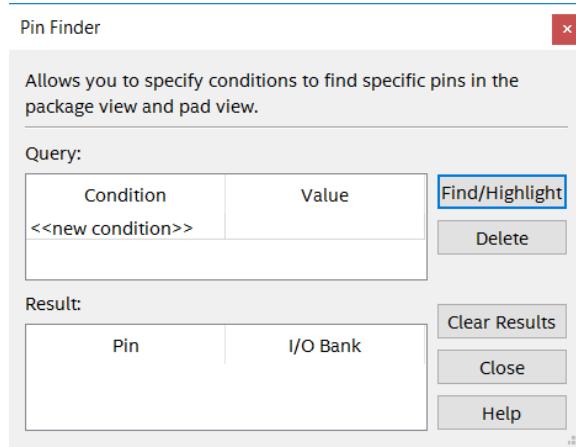


More Pin Planner Features (1)

- Pin Finder

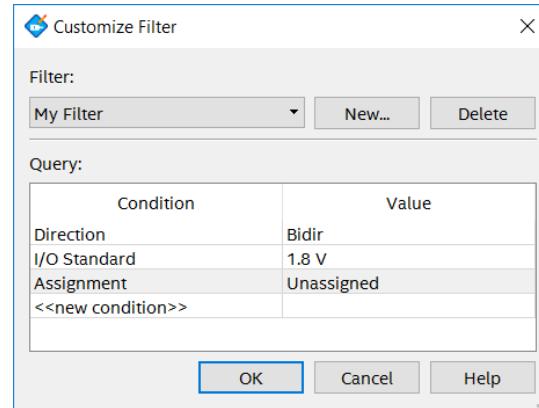
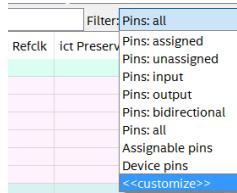


- Locate pins meeting user-defined criteria
- Use to find compatible pin locations
- Pins highlighted in Package View



- Custom Filters

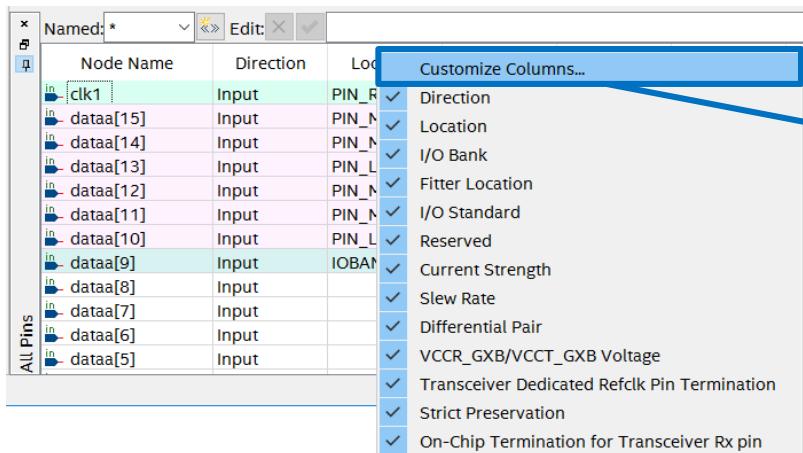
- Create custom filters for All Pins list



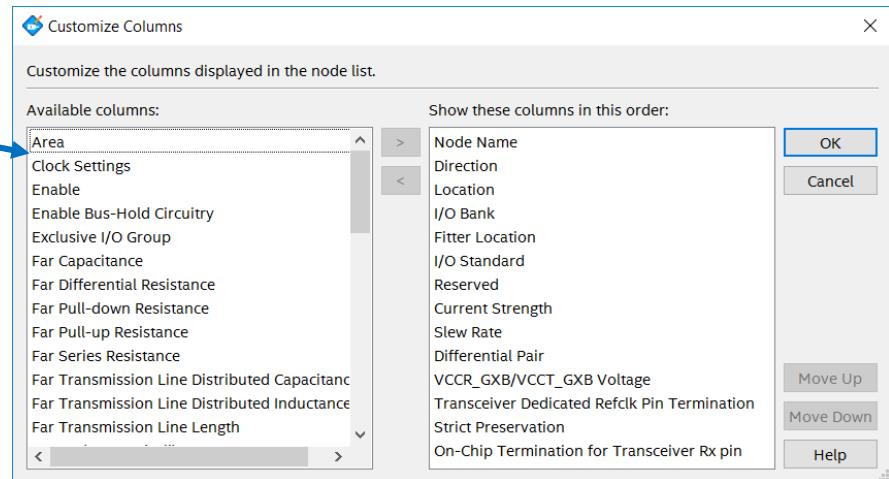
More Pin Planner Features (2)

■ Customizable columns

- Select the I/O-related assignment columns to be shown in All Pins list for easy management
- Right-click on column heading to select which columns will be displayed



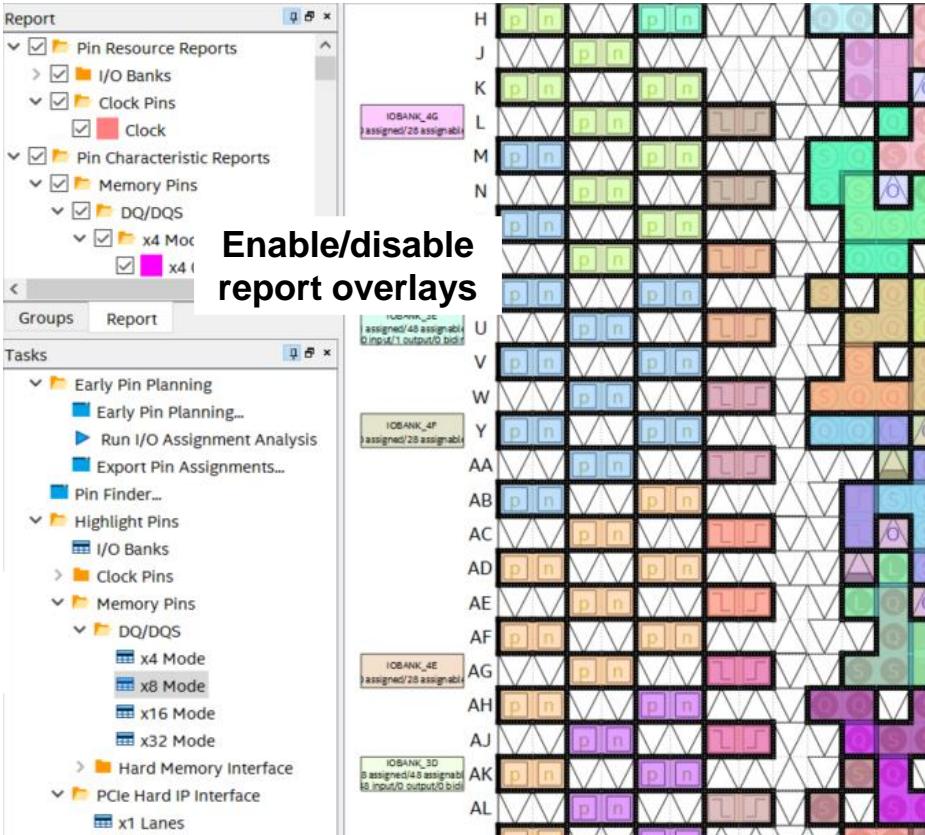
Node Name	Direction	Loc	Customize Columns...
in clk1	Input	PIN_R	<input checked="" type="checkbox"/> Direction
in dataa[15]	Input	PIN_M	<input checked="" type="checkbox"/> Location
in dataa[14]	Input	PIN_N	<input checked="" type="checkbox"/> I/O Bank
in dataa[13]	Input	PIN_L	<input checked="" type="checkbox"/> Fitter Location
in dataa[12]	Input	PIN_N	<input checked="" type="checkbox"/> I/O Standard
in dataa[11]	Input	PIN_N	<input checked="" type="checkbox"/> Reserved
in dataa[10]	Input	PIN_L	<input checked="" type="checkbox"/> Current Strength
in dataa[9]	Input	IOBAN	<input checked="" type="checkbox"/> Slew Rate
in dataa[8]	Input		<input checked="" type="checkbox"/> Differential Pair
in dataa[7]	Input		<input checked="" type="checkbox"/> VCCR_GXB/VCCT_GXB Voltage
in dataa[6]	Input		<input checked="" type="checkbox"/> Transceiver Dedicated Refclk Pin Termination
in dataa[5]	Input		<input checked="" type="checkbox"/> Strict Preservation
			<input checked="" type="checkbox"/> On-Chip Termination for Transceiver Rx pin



Pin Planner Tasks & Report Windows

- Use **Tasks** window to access common tasks
- **Highlight Pins** tasks will generate colored overlays similar to the Chip Planner
 - Enable/disable reports in **Report** pane

Double-click to perform any task



Reserved and Unused I/O Pins

- Type reserved I/O name into All Pins list & select reserved setting
- Prevents Fitter from placing unassigned signal on pin (discussed next)

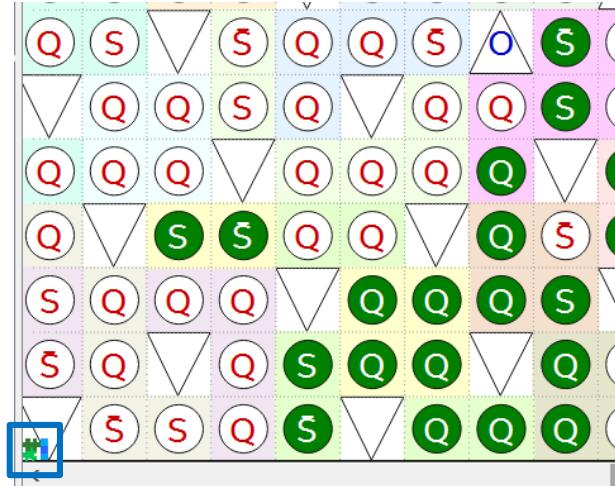
Node Name	Direction	Location	I/O Bank	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differen
dataa[10]	Input	PIN_L18	3G	PIN_L18	1.8 V				
dataa[11]	Input	PIN_M18	3G	PIN_M18	1.8 V				
dataa[12]	Input	PIN_M17	3G	PIN_M17	1.8 V				
dataa[13]	Input	PIN_L15	3G	PIN_L15	1.8 V				
dataa[14]	Input	PIN_M15	3G	PIN_M15	1.8 V				
dataa[15]	Input	PIN_M16	3G	PIN_M16	1.8 V				
clk1	Input	PIN_R14	3E	PIN_R14	1.8 V				
my_reserved_pin	Input				1.8 V (default)	As bidirectional As input tri-stated As output driving VCC As output driving an unspecified signal As output driving ground			
<<new node>>									

- Or right-click on pin in Package View and click Reserve Pins → As...
 - Pin name set to user_reserve_<number>
- Set initial state of other unused pins in Device settings dialog box (shown later)

Show Fitter Placements

- View I/O locations selected by the Fitter

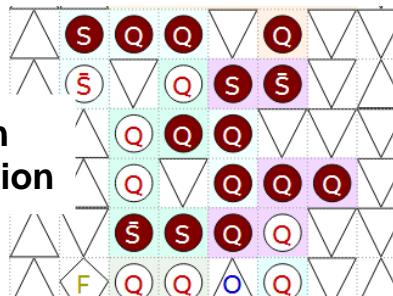
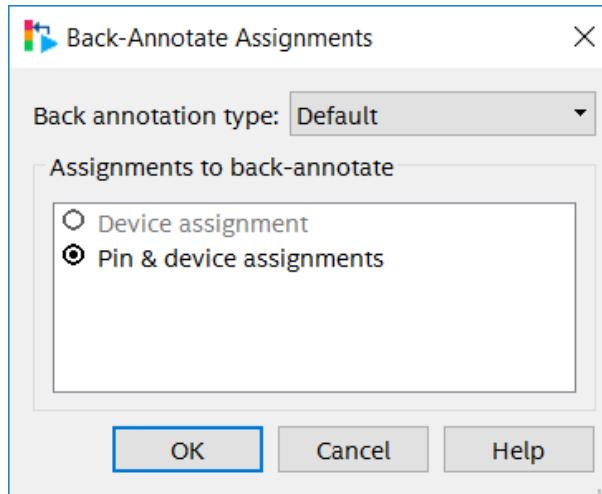
View → Show
or in toolbar



Back-Annotation

- Use to lock Fitter-chosen pin assignments for future compilations
 - Copies device & resource locations chosen by fitter into **.qsf** file
 - Pins
 - Logic
 - Routing
- “Locks down” locations in Pin Planner

Assignments menu in main Intel® Quartus® Prime Design Software window



Green/brown pattern
indicates back-annotation



The Intel® Quartus® Prime Design Software: Foundation

Verify I/O Assignments with the Pin Planner

Verifying I/O Assignments

Run from Pin Planner toolbar

- I/O Assignment Analysis
 - Checks legality of all I/O assignments without full compilation
- Minimal requirements for running
 - I/O declaration
 - HDL port declaration
 - Reserved pin
 - Pin-related assignments
 - I/O standard
 - Current strength
 - Pin location (pin, bank, edge)
 - Toggle rate



Pin Planner Processing menu → Start
I/O Assignment Analysis
or toolbar

I/O Rules Checked

- No internal logic
 - Checks I/O locations & constraints with respect to other I/O & I/O banks
 - e.g. Each I/O bank supports a single VCCIO
- I/O connected to logic
 - Checks I/O locations & constraints with respect to other I/O, I/O banks, & internal resources
 - e.g. PLL must be driven by a dedicated clock input pin
- Note: When working with design files, synthesize design before running I/O Assignment Analysis

I/O Assignment Analysis Output

The screenshot shows the Quartus Prime Compilation Report interface. The left sidebar contains a 'Table of Contents' with sections like Flow OS Summary, Flow Log, Synthesis, Fitter, Parallel Compilation, Partition Summary, Netlist Optimizations, Plan Stage, Device Options, Operating Settings and Conditions, Pin-Out File, Input Pins (which is selected and highlighted in blue), Output Pins, I/O Bank Usage, All Package Pins, I/O Assignment Warnings, Control Signals, Global & Other Fast Signals Summary, Global & Other Fast Signals Details, and Plan Messages. The main panel is titled 'Input Pins' and displays a table with 19 rows of data:

	Name	Pin #	I/O Bank	X coordinate	Y coordinate	Z coon
1	clk1	R14	3E	142	115	46
2	dataa[0]	BC23	3A	142	12	31
3	dataa[10]	L18	3G	142	175	16
4	dataa[11]	M18	3G	142	176	16
5	dataa[12]	M17	3G	142	172	16
6	dataa[13]	L15	3G	142	170	16
7	dataa[14]	M15	3G	142	169	16
8	dataa[15]	M16	3G	142	171	16
9	dataa[1]	AR25	3A	142	15	61
10	dataa[2]	BC25	3A	142	14	31
11	dataa[3]	AU22	3A	142	7	61
12	dataa[4]	BA18	3A	142	11	16
13	dataa[5]	BB22	3A	142	7	31
14	dataa[6]	BA22	3A	142	6	31
15	dataa[7]	AP24				
16	dataa[8]	BD22				
17	dataa[9]	G32				
18	datab[0]	BC20				
19	datab[10]	AU25				

Messages on I/O assignment issues

- Compiler assumptions
- Device & pin migration issues
- I/O bank voltages & standards

Validating I/O Pin-out

- Completed design
 - Run full compilation
 - Enable option to **Run I/O Assignment Analysis before compilation (Settings dialog box → Compilation Process Settings)**
- Incomplete design with completed top-level design file
 - Add I/O-related IP cores and instantiate, even if final connections are not complete
 - Run I/O Assignment Analysis on design

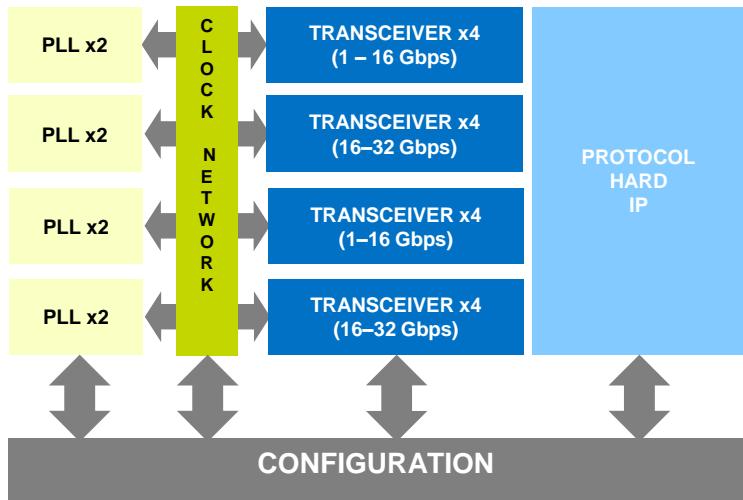


The Intel® Quartus® Prime Design Software: Foundation

I/O Planning with the Interface Planner

Problem: I/O Interfaces are Complex

- FPGAs support hundreds of I/O protocols, dozens of memory standards
- Protocols have thousands of configurations specified by standards
- Example rules
 - Neighbor I/O to 28 Gbps I/O restricted to 14 Gbps
 - PCIe* master channel restricted to specific transceiver lanes
 - DDR3-144 can only exist in Banks A to C if PLL #21 needs to be used by fabric logic



Problem: Periphery Placement Rules are Complex



Input design with
illegal or incomplete
assignments

Intel® Quartus® Prime
Design Software

No fit!!

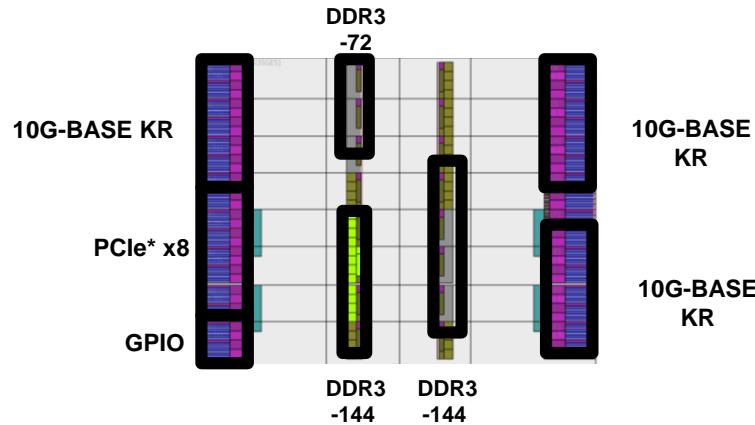
Error (175022): The auto-promoted clock driver could not be placed in any location to satisfy its connectivity requirements

???

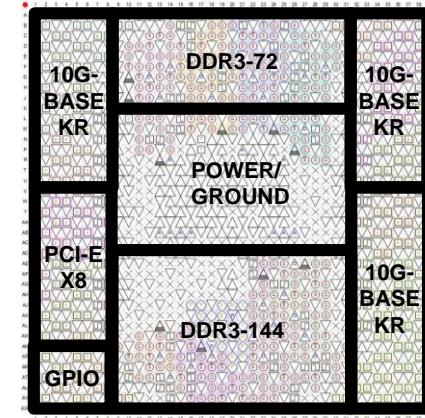
Better Solution: Interfaced-Based Assignment

- Instead of pin-by-pin or bus-by-bus, interface-by-interface!
- Choose valid locations for interfaces and clock domains graphically
- Verify assignments and update valid locations as assignments are made

By device resources

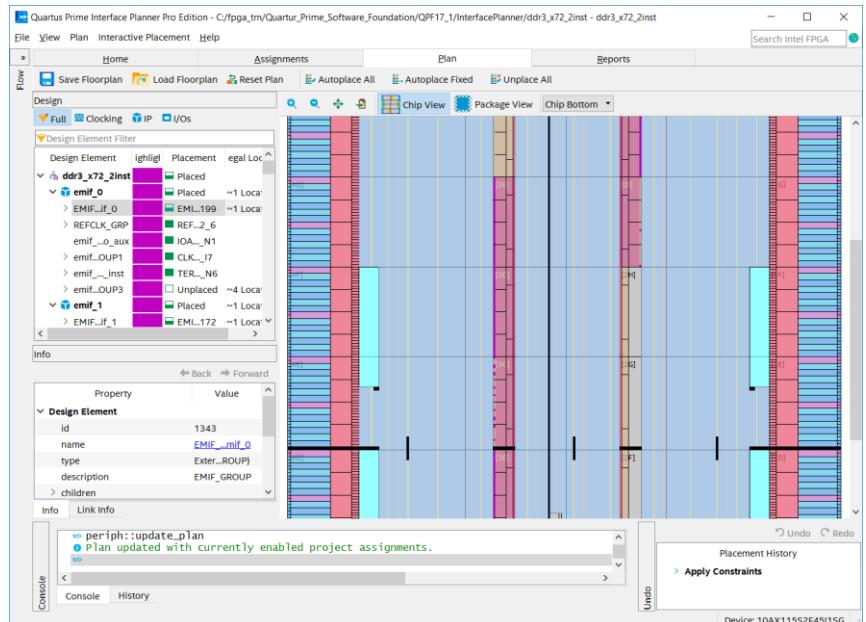


By I/O pins



Introducing the Interface Planner

- Create legal assignments using the power of the Fitter
- Verify and validate in real-time
- High impact productivity tool for faster design closure
- Legal mapping of package footprint to I/O protocols within minutes
- Develop board layouts with confidence
- Available only in the Intel® Quartus® Prime Pro Edition software
- For Intel® Arria® 10 and newer devices



Plan Tab Chip View

- Click and drag post-synthesis design elements (parts of design requiring placement in device) to the **Chip View**



List Available Locations

Design

Full Clocking IP I/Os

Design Element Filter

Design Element	Highlight	Placement	Legal Locations
ddr3_x72_2inst	Placed		
emif_0	Unplaced	>10 Locations >>	
emif_1	Unplaced	>10 Locations >>	
ALTER..._AUX0	Unplaced	~4 Locations >>	
ALTER..._AUX1	Unplaced	~4 Locations >>	
ALTER..._AUX2	Unplaced	4 Locations >>	
ALTER..._AUX3	Unplaced	~4 Locations >>	

Chip View Package View Chip Bottom

Legal Locations

Filter

Location

Legal Locations for emif_0

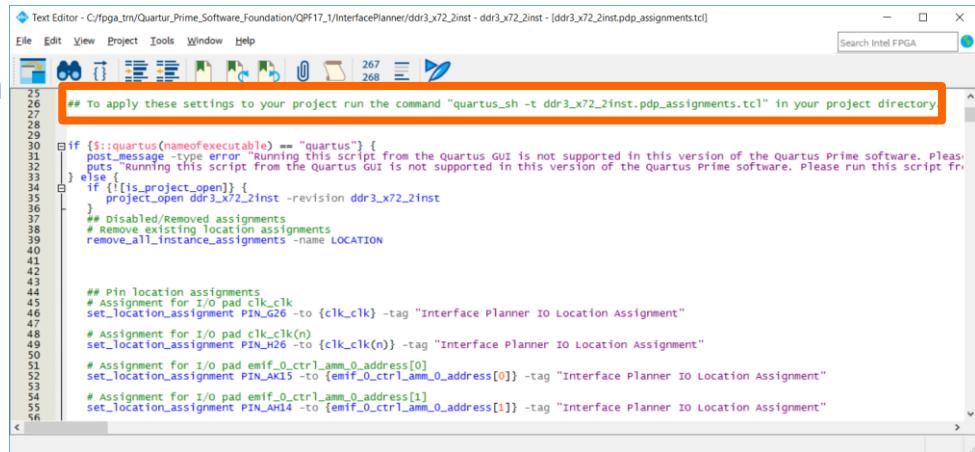
- EMIF_L4_X78_Y196_Y199
- EMIF_L4_X78_Y169_Y172
- EMIF_L4_X78_Y142_Y145
- EMIF_L4_X78_Y115_Y118
- EMIF_L4_X142_Y196_Y199
- EMIF_L4_X142_Y169_Y172
- EMIF_L4_X142_Y142_Y145
- EMIF_L4_X142_Y115_Y118
- EMIF_L4_X142_Y88_Y91
- EMIF_L4_X142_Y61_Y64
- EMIF_L4_X142_Y33_Y36
- EMIF_L4_X142_Y6_Y9

Click arrows to highlight locations without drag and drop

Click location to highlight or double-click to assign design element

Source Assignments Into Project & Compile

- Source generated Tcl script with command line or **Tcl Scripts** command in **Tools** menu
- Script provides command for sourcing
- Script organized into sections; pick and choose types of assignments to use



The screenshot shows a Windows-style text editor window with a toolbar at the top. The title bar reads "Text Editor - C:/fpga_tm/Quartus_Prime_Software/Foundation/QPF17_1/InterfacePlanner/ddr3_x72_2inst - ddr3_x72_2inst.pdp_assignments.tcl". The menu bar includes File, Edit, View, Project, Tools, Window, and Help. The right side of the window has a vertical scrollbar. The main area contains a block of Tcl script code.

```
25  # To apply these settings to your project run the command "quartus_sh -t ddr3_x72_2inst.pdp_assignments.tcl" in your project directory
26
27
28
29
30  if ${!:quartus(nameofexecutable)} == "quartus" {
31    print_message -type error "Running this script from the Quartus GUI is not supported in this version of the Quartus Prime software. Please run this script from the command line"
32    puts "Running this script from the Quartus GUI is not supported in this version of the Quartus Prime software. Please run this script from the command line"
33  } else {
34    if {[is_project_open]} {
35      project_open ddr3_x72_2inst -revision ddr3_x72_2inst
36    }
37    ## Disabled/Removed assignments
38    ## Remove existing location assignments
39    remove_all_instance_assignments -name LOCATION
40
41
42
43
44    ## Pin location assignments
45    # Assignment for I/O pad clk_clk
46    set_location_assignment PIN_G26 -to {clk_clk} -tag "Interface Planner IO Location Assignment"
47    # Assignment for I/O pad clk_clk(n)
48    set_location_assignment PIN_H26 -to {clk_clk(n)} -tag "Interface Planner IO Location Assignment"
49    # Assignment for I/O pad emif_0_ctrl_amm_0_address[0]
50    set_location_assignment PIN_A15 -to {emif_0_ctrl_amm_0_address[0]} -tag "Interface Planner IO Location Assignment"
51    # Assignment for I/O pad emif_0_ctrl_amm_0_address[1]
52    set_location_assignment PIN_A14 -to {emif_0_ctrl_amm_0_address[1]} -tag "Interface Planner IO Location Assignment"
```



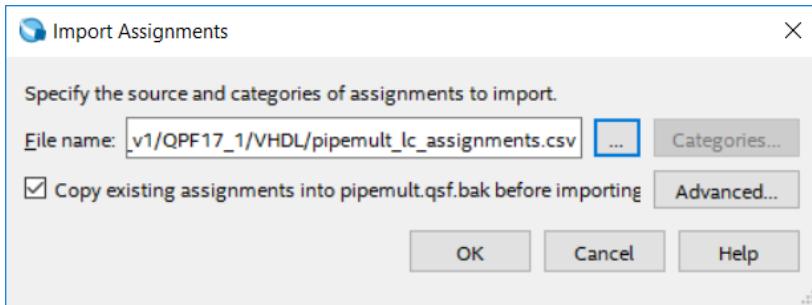
The Intel® Quartus® Prime Design Software: Foundation

Other Methods of Creating I/O-Related Assignments

Import/Export via CSV

- Use spreadsheet Comma Separated Value (.csv) file to enter or edit I/O locations
- Convenient for transferring assignments between project revisions
- CSV column names must match Pin Planner column headings

Pin Planner **File** menu or Intel® Quartus® Prime Design Software **Assignments** menu



	A	B	C	D	E	F
25	dataaa[13]	Input	PIN_J24	3H	PIN_A19	1.8 V
26	dataaa[12]	Input	PIN_E25	3H	PIN_D19	1.8 V
27	dataaa[11]	Input	PIN_L25	3H	PIN_B20	1.8 V
28	dataaa[10]	Input	PIN_B25	3H	PIN_K26	1.8 V
29	dataaa[9]	Input	PIN_D19	3H	PIN_M23	1.8 V
30	dataaa[8]	Input	PIN_A20	3H	PIN_B23	1.8 V

.qsf Editing & Scripting

- Type pin-related assignments directly into .qsf
- Type pin-related assignments into separate Tcl
 - Source Tcl file in project .qsf
 - Execute Tcl file to write assignments into .qsf

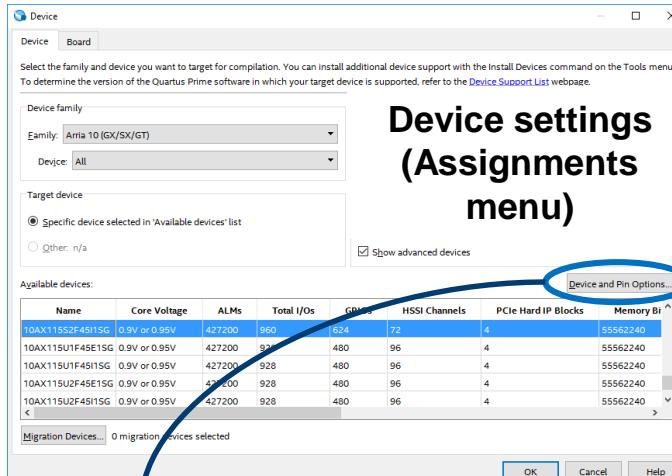
```
# file is updated automatically by the Quartus Prime software
# and any changes you make may be lost or overwritten.
#
# -----
37 source "location_assignments.tcl"
38
39
40 set_global_assignment -name FAMILY "Arria 10"
41 set_global_assignment -name DEVICE 10AX115S2F45I1SG
42 set_global_assignment -name TOP_LEVEL_ENTITY pipemult
43 set_global_assignment -name ORIGINAL_QUARTUS_VERSION 17.1.0
44 set_global_assignment -name PROJECT_CREATION_TIME_DATE "13:57:03 08
45 set_global_assignment -name LAST_QUARTUS_VERSION "17.1.0 Pro Edition
46 set_global_assignment -name PROJECT_OUTPUT_DIRECTORY output_files
47
Ln 46 Col 1          Quartus Prime Setting File
```



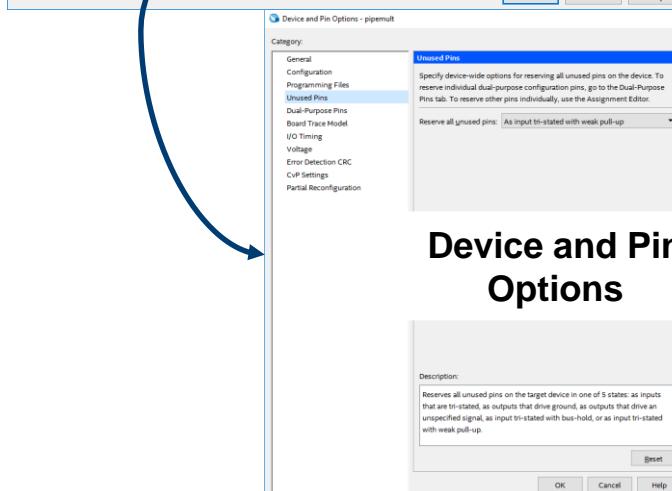
```
22 set_instance_assignment -name IO_STANDARD "1.8 V" -to dataa
23 set_location_assignment IOBANK_3H -to dataa
24 set_location_assignment IOBANK_3G -to q
25 set_instance_assignment -name IO_STANDARD "1.5 V" -to q
26 set_location_assignment IOBANK_3G -to rdaddress
27 set_location_assignment IOBANK_3G -to wraddress
28 set_instance_assignment -name IO_STANDARD "1.5 V" -to rdaddress
29 set_instance_assignment -name IO_STANDARD "1.5 V" -to wraddress
30 set_location_assignment PIN_G22 -to dataa[15]
31 set_location_assignment PIN_C23 -to clk1
32 set_location_assignment PIN_H20 -to wren
33 set_instance_assignment -name IO_STANDARD "1.8 V" -to dataa[15]
34 set_instance_assignment -name IO_STANDARD "1.8 V" -to dataa[14]
35 set_instance_assignment -name IO_STANDARD "1.8 V" -to dataa[13]
```

Global Pin Settings

- Select global settings for pins
 - e.g. unused I/O configuration, dual-purpose configuration pins, device CRC checking
- Assignments in Pin Planner or other tools have precedence over global settings



Device settings (Assignments menu)



Device and Pin Options

I/O Planning Summary

- Pin assignments can be performed in many ways, graphically & by means of text files
- The Pin Planner provides an easy-to-use graphical method for creating and managing pin assignments
- The Interface Planner combines I/O interface resource assignment with on-the-fly legality checking
- I/O Assignment Analysis helps validate a device pin-out without performing a full compilation
- Pin validation can be completed during any point in design development

I/O Planning Support Resources

- Intel® Quartus® Prime Design Software Handbook chapters (all Volume 2)
 - Managing Device I/O Pins
 - Interface Planner Design Planning
 - Reviewing Printed Circuit Board Schematics with the Intel® Quartus® Prime Software
 - Signal Integrity Analysis with Third-Party Tools
 - Mentor Graphics* PCB Design Tools Support
 - Cadence* PCB Design Tools Support
- Training courses and demonstrations
 - [Fast & Easy I/O System Design with Interface Planner](#)
 - [I/O Signal Integrity Analysis with Third-Party Tools](#)



The Intel® Quartus® Prime Design Software: Foundation

A Taste of Programming & Configuration

Programming Files (1)

.sof (SRAM Object File)

- Used to configure FPGAs directly from Intel® Quartus® Prime Design Software through download cable
- Always generated by default during a full compilation by the Assembler

.pof (Programming Object File)

- Used to program CPLDs and configuration devices

.jam/.jbc

- ASCII file used by processors and test equipment to program devices via JTAG

Programming Files(2)

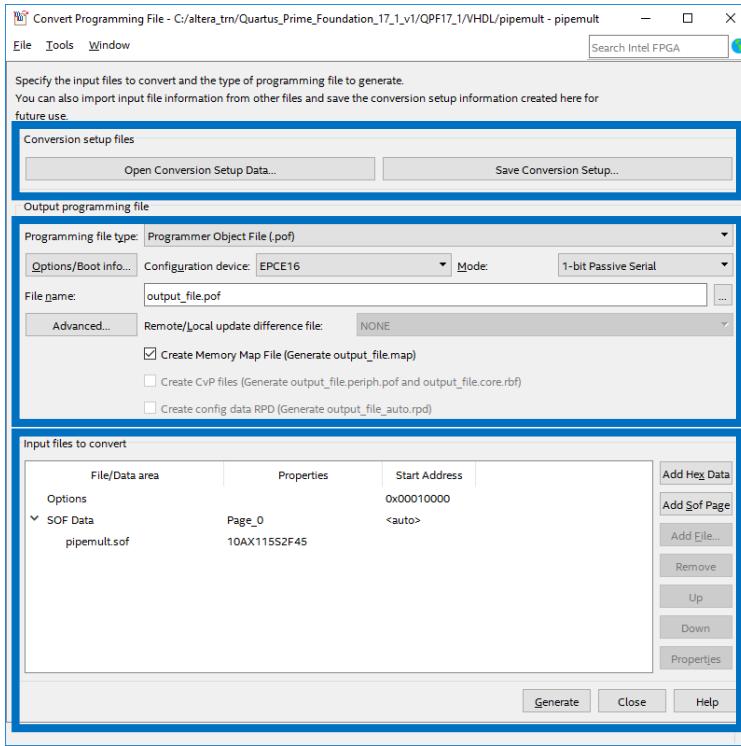
.jic (JTAG Indirect Configuration File)

- Contains programming data for target FPGA
- Used to program EPICS (Intel FPGA serial configuration) devices through their dedicated configuration interface connection to an FPGA over FPGA's JTAG interface
- Configuration devices do not have JTAG interfaces!

Programming File Conversion

- Intel® Quartus® Prime Software Assembler by default automatically generates **.sof** files for any FPGA
 - May generate additional single-device programming file types from Device settings dialog box (**Device and Pin Options → Programming Files**)
- **.sof** files can only be used by the Intel® Quartus® Prime Software Programmer to directly configure FPGA
- Other configuration solutions may require converting **.sof** file(s) into other file formats

Convert Programming Files GUI



File menu → Convert Programming Files

Load or save file conversion setup (.cof)

Select an optional programming file format

Select a configuration device and its supported mode

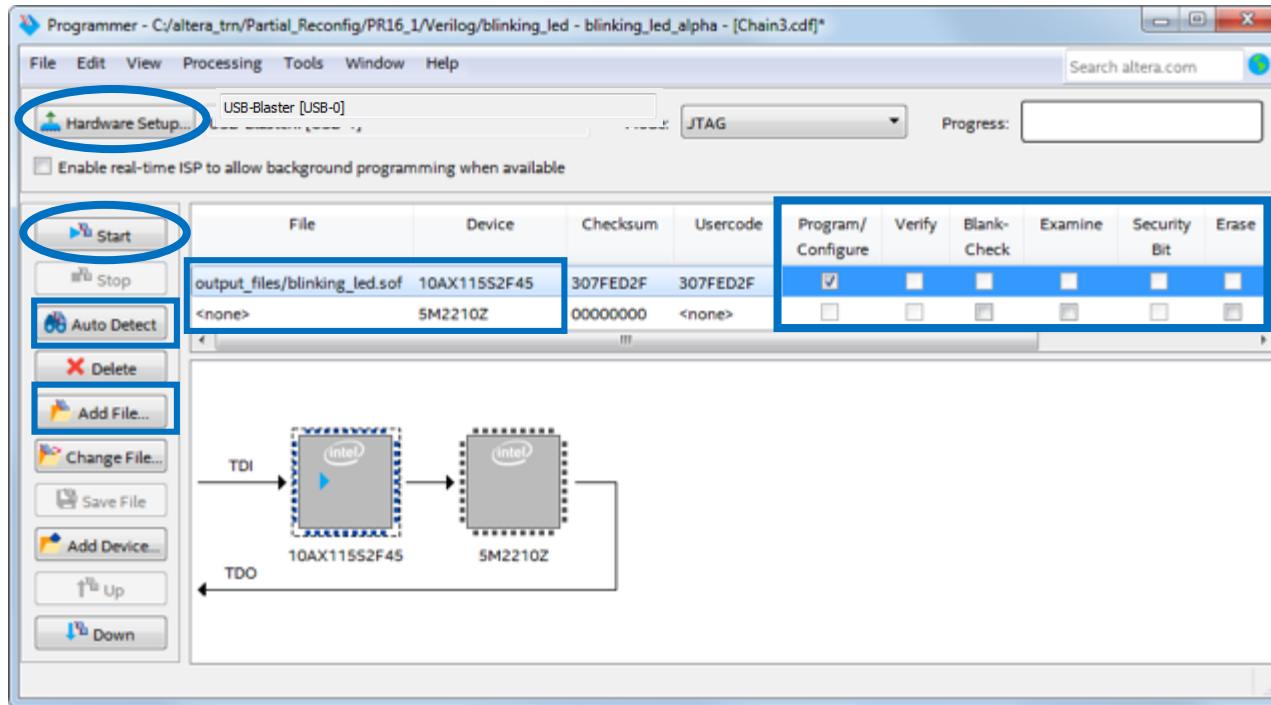
Select more options for the configuration device

Add input configuration/programming file(s)

Intel® Quartus® Prime Software Programmer



Tools menu → Programmer



Programming/Configuration Support Resources

- Intel® Quartus® Prime Design Software Handbook chapter
 - Programming Intel® FPGA Devices (Volume 3)
- Training & Demonstrations
 - [Introduction to Configuring Intel FPGAs](#)
 - [Configuration Schemes for Intel FPGAs](#)
 - [Configuration Solutions for Intel FPGAs](#)
 - [Debugging JTAG Chain Integrity](#)

Exercise 5

Class Summary

- Intel® Quartus® Prime Design Software Projects
- Design Entry
- Intel Quartus Prime Software Compilation
- Settings & Assignments
- I/O Planning
- Programming/Configuration

Additional Intel® Quartus® Prime Design Software Instructor-Led Courses (1)

- Intel® Quartus® Prime Software Debug and Analysis Tools

- Debugging solutions
 - Signal Probe incremental routing
 - Signal Tap Embedded Logic Analyzer
 - In-System Sources & Probes
 - In-System Memory Content Editor
 - Chip Planner & Resource Property Editor
 - System Console and related toolkits

- Intel® Quartus® Prime Design Software Series: Timing Analysis

- Create timing constraints to meet and optimize timing using the Timing Analyzer
 - Perform detailed timing analysis on an Intel® FPGA device with the Timing Analyzer

Additional Intel® Quartus® Prime Design Software Instructor-Led Courses (2)

- Advanced Timing Analysis with Timing Analyzer
 - Automate constraining and analysis of FPGA designs using Tcl
 - Constrain advanced types of interfaces and blocks
- Timing Closure with the Intel® Quartus® Prime Software
 - Employ best practices to close timing using the tools available
- Introduction to the Platform Designer
 - Integrate IP and custom logic into a Platform Designer system

Many Ways to Learn!



Videos

FREE
Always available
~4 minutes long
YouTube videos



Online Training

FREE
Always available
~30 minutes long
>200 topics
English, Chinese, Japanese



Virtual Classes

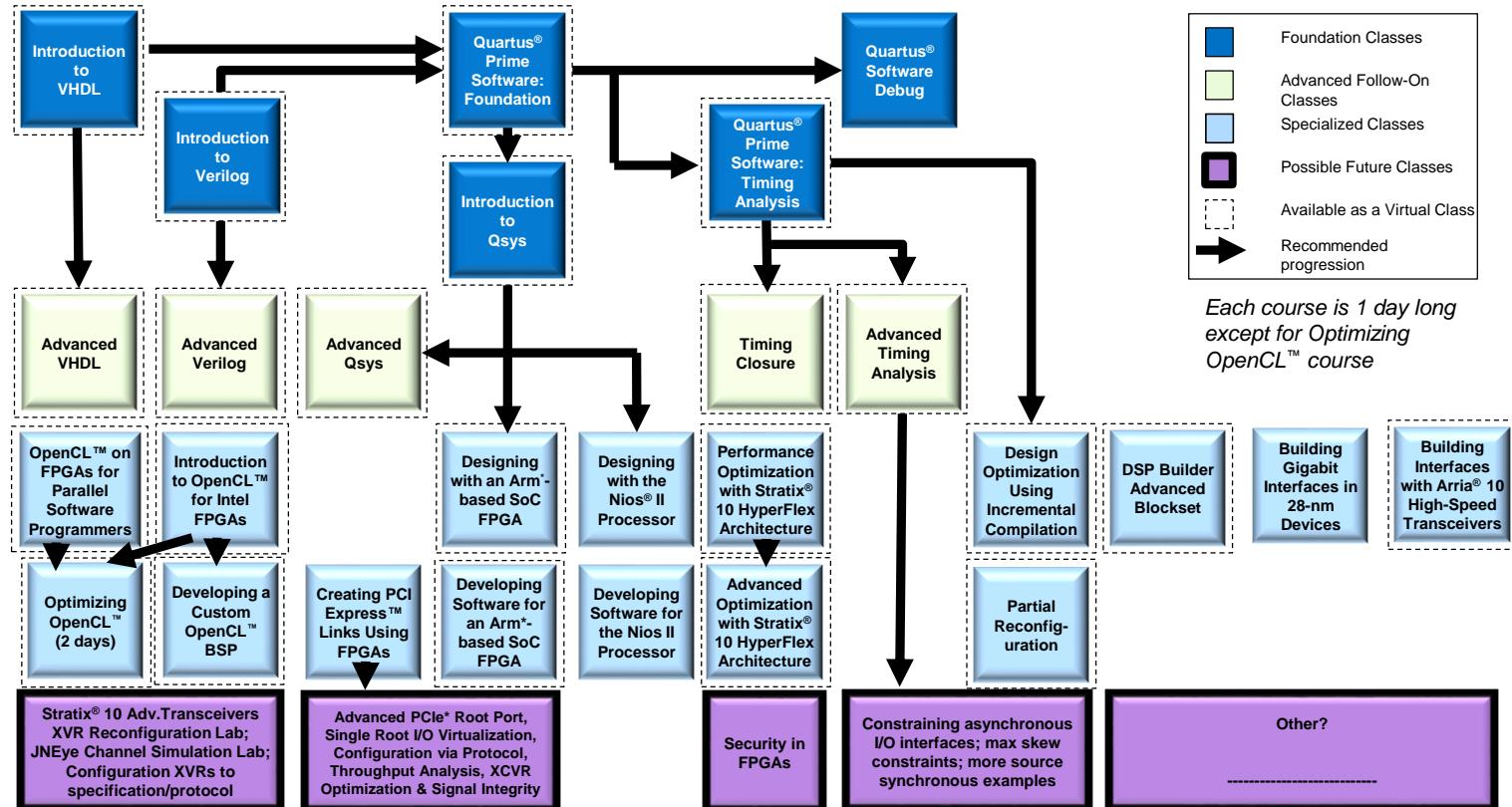
Live over Webex
Ask questions to Intel®
FPGA expert
Hands on labs
Taught in ½ day sessions
Class schedules at
www.altera.com/training



Instructor-led Training

In-person
Ask questions to
Intel® FPGA expert
Hands on labs
1 day long
Class schedules at
www.altera.com/training

Instructor-Led and Virtual Training Curriculum



Intel® FPGA Technical Support Resources

SUPPORT

- Intel® FPGA [Technical Training](#) materials
- Intel ® Programmable Solutions Group (PSG) [community forum](#) for self-help
- Intel PSG [wiki site](#) for design examples
- Intel PSG Knowledge Base [Solutions](#)
- Intel PSG [Self Servicing License Center](#)
- Please contact your sales and field support if you need further assistance

Give Us Your Feedback

When you registered for this training you received a confirmation email

Please click on the link in the email to complete a short survey

Your feedback is important to help us improve future trainings!

Legal Disclaimers/Acknowledgements

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. Check with your system manufacturer or retailer or learn more at www.intel.com.

Intel, the Intel logo, Intel Inside, the Intel Inside logo, MAX, Stratix, Cyclone, Arria, Quartus, HyperFlex, Intel Atom, Intel Xeon and Enpirion are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

OpenCL is the trademark of Apple Inc. used by permission by Khronos

*Other names and brands may be claimed as the property of others

© Intel Corporation

