

Quartus II Software Design Series: Foundation

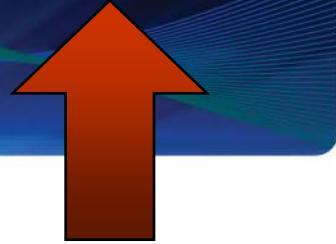
Online Training



© 2013 Altera Corporation—Public



Supplemental Files to Download



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

Course Objectives

- **Create a new Quartus® II project**
- **Choose supported design entry methods**
- **Compile a design into a PLD**
- **Locate resulting compilation information**
- **Create design constraints (assignments & settings)**
- **Manage I/O assignments**
- **Program/configure a PLD**

Class Agenda

- Intro to Altera® & Devices
- Quartus II Feature Overview
- Projects
 - Exercise 1
- Design Methodology
- Design Entry
 - Exercise 2
- Compilation
 - Exercise 3
- Settings & Assignments
 - Exercise 4
- I/O Planning
 - Exercise 5
- Programming/Configuration

Quartus II Software Design Series: Foundation

Introduction to Altera & Altera Devices



© 2013 Altera Corporation—Public



A Complete Solutions Portfolio

ALTERA®



CPLDs



Low-cost FPGAs



High-density,
high-performance FPGAs



Mid-range Transceiver
FPGAs



ASICs

Nios® II

Embedded
soft processors



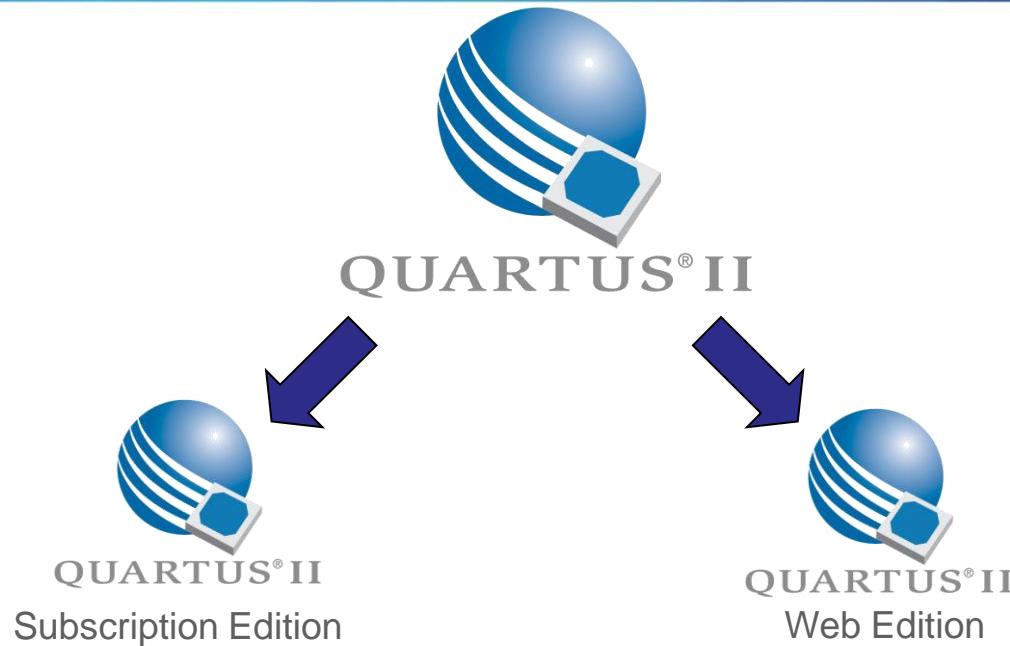
Intellectual
Property (IP)



Development
kits

ALTERA®
MEASURABLE ADVANTAGE™

Quartus II Software – Two Editions



Devices Supported	All	Selected Devices
Features	100%	95%
Distribution	Internet & DVD	Internet & DVD
Price	Paid	Free (no license required)

[Feature Comparison available on Altera web site](#)

Quartus II Software Design Series: Foundation

Quartus II Design Software Feature Overview



© 2013 Altera Corporation—Public



■ Fully-integrated development tool

- Multiple design entry methods
- Logic synthesis
- Place & route
- Static timing analysis
- Power and SSN analysis
- Device programming

■ Simulation

- Supports standard HDL simulation tools
- Includes ModelSim®-Altera Starter Edition tool
 - Optional upgrade to ModelSim-Altera Edition tool

More Features

- **MegaWizard® Plug-In Manager & Qsys design tools**
- **TimeQuest Timing Analyzer**
- **Incremental compilation feature**
- **PowerPlay Power Analyzer**
- **NativeLink 3rd-party EDA tool integration**
- **Debugging capabilities - From HDL to device in-system**
- **32 & 64-bit Windows & Linux support**
- **Multi-processor support**
- **Node-locked & network licensing options**

Welcome to the Quartus II Software!



The image shows the Quartus II Software welcome screen. It features a blue and white design with a 3D sphere icon. The text "QUARTUS® II" is displayed. The screen is divided into two main sections: "Start Designing" on the left and "Start Learning" on the right. A yellow callout box on the right side contains the text "Turn on or off in Tools → Options".

Getting Started With Quartus® II Software

Start Designing

Designing with Quartus II software requires a project

Create a New Project (New Project Wizard)

Open Existing Project

Open Recent Project:

- C:/altera_trn/VER/lab4a/reg16.qpf
- C:/altera_trn/VER/lab4b/counter.qpf
- C:/altera_trn/VER/lab5a/mult_control.qpf
- C:/altera_trn/VER/lab5b/mult8x8.qpf

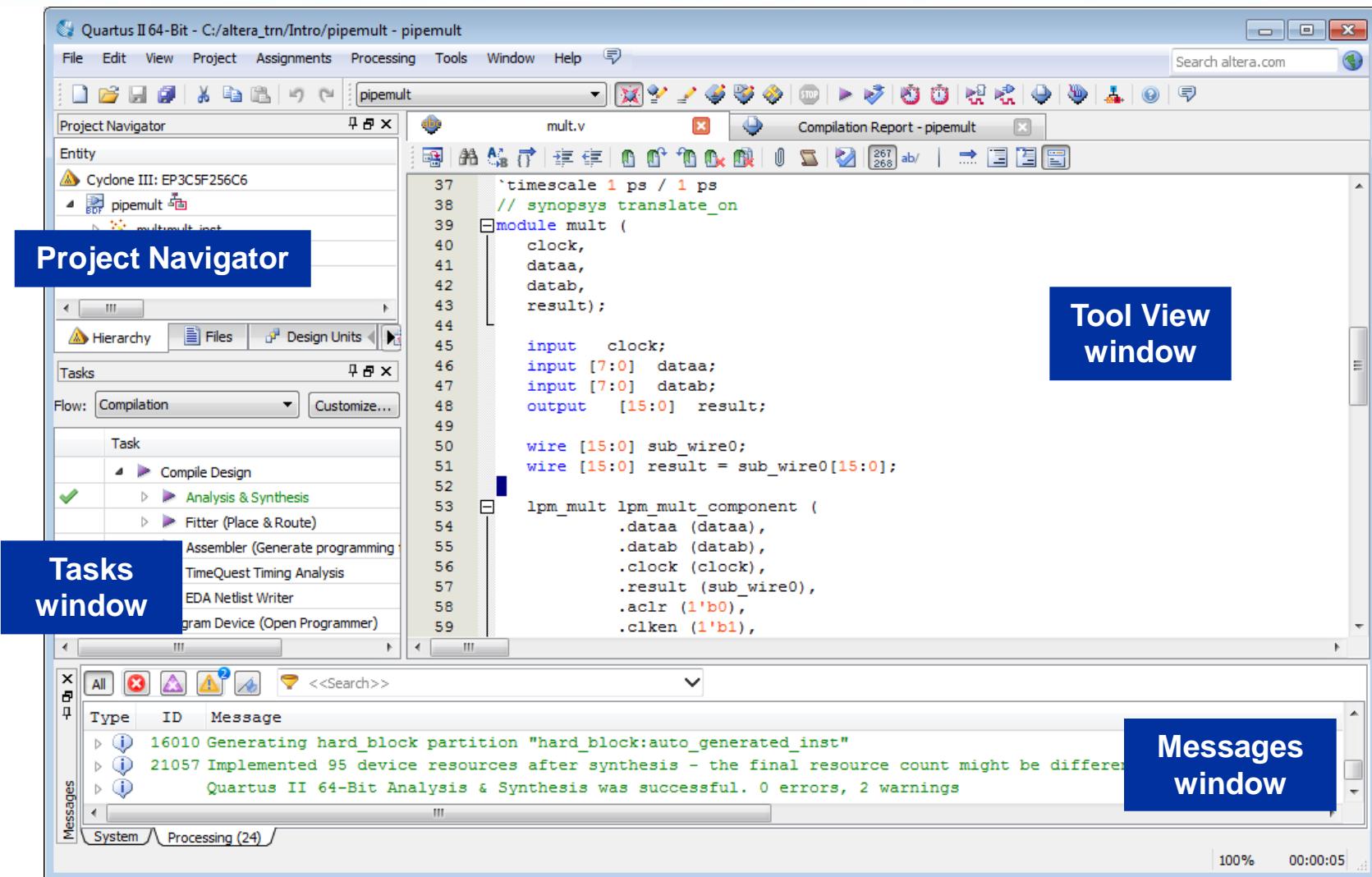
Turn on or off in Tools → Options

Don't show this screen again

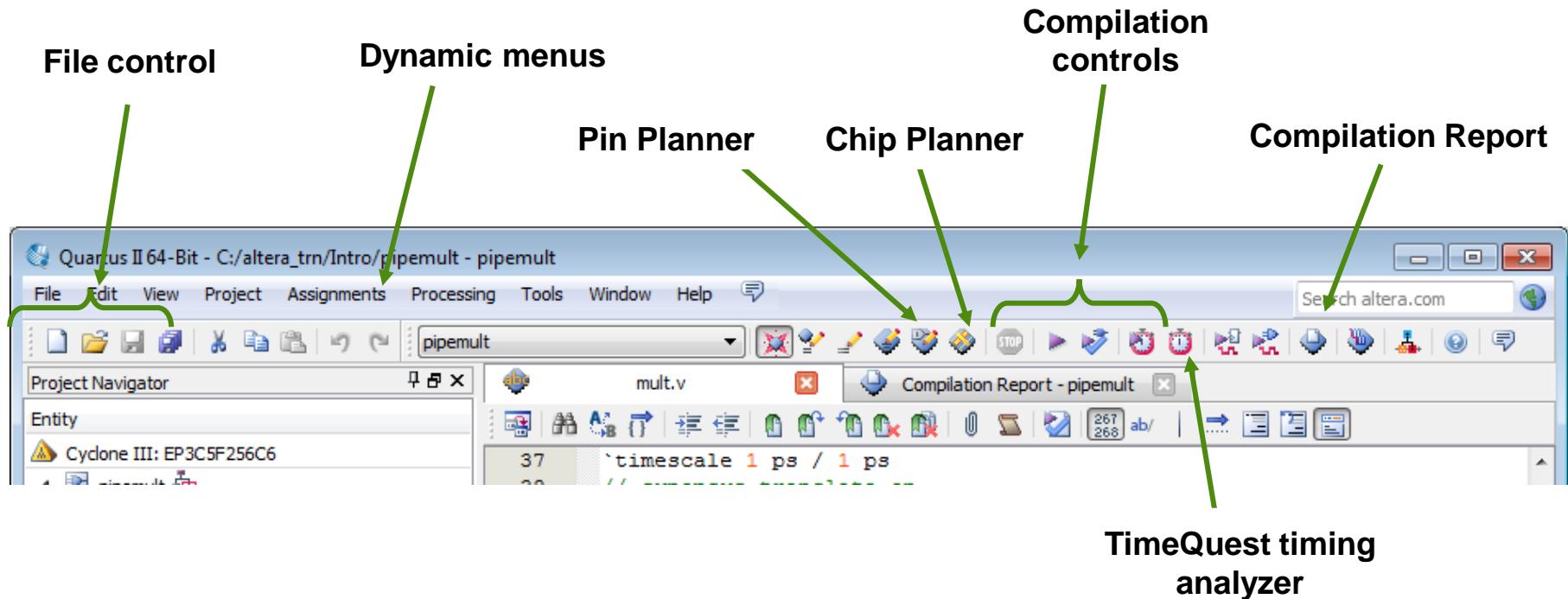
Literature **Training** **Online Demos** **Support**

ALTERA

Quartus II Default Operating Environment



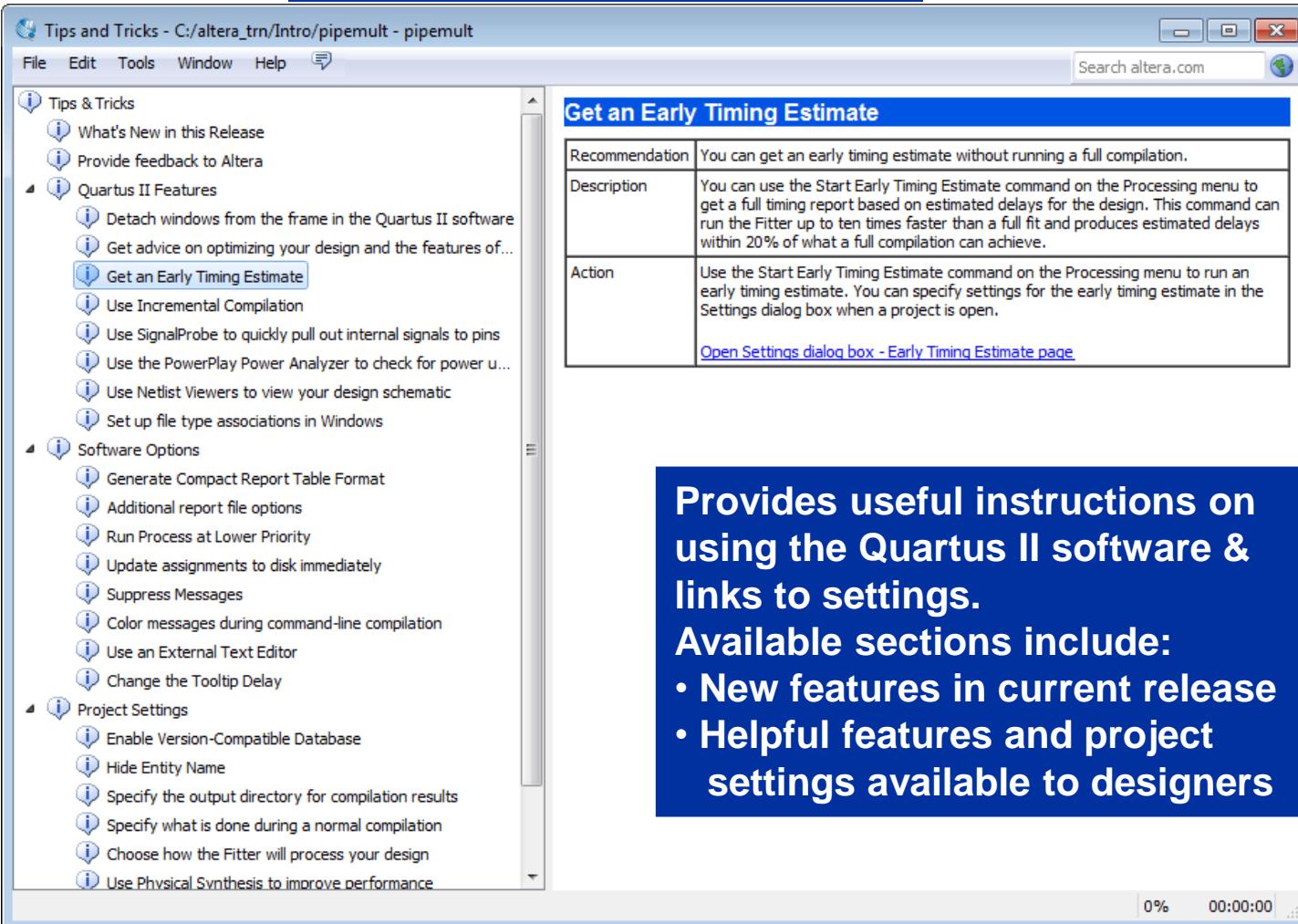
Main Toolbar



To Customize Toolbars
Tools → Customize...

Tips & Tricks Advisor

Help menu → Tips & Tricks



The screenshot shows the 'Tips and Tricks' window for a project named 'C:/altera_trn/Intro/pipemult - pipemult'. The window has a sidebar with a tree view of tips, and the main area displays a specific tip titled 'Get an Early Timing Estimate'.

Get an Early Timing Estimate

Recommendation	You can get an early timing estimate without running a full compilation.
Description	You can use the Start Early Timing Estimate command on the Processing menu to get a full timing report based on estimated delays for the design. This command can run the Fitter up to ten times faster than a full fit and produces estimated delays within 20% of what a full compilation can achieve.
Action	Use the Start Early Timing Estimate command on the Processing menu to run an early timing estimate. You can specify settings for the early timing estimate in the Settings dialog box when a project is open.

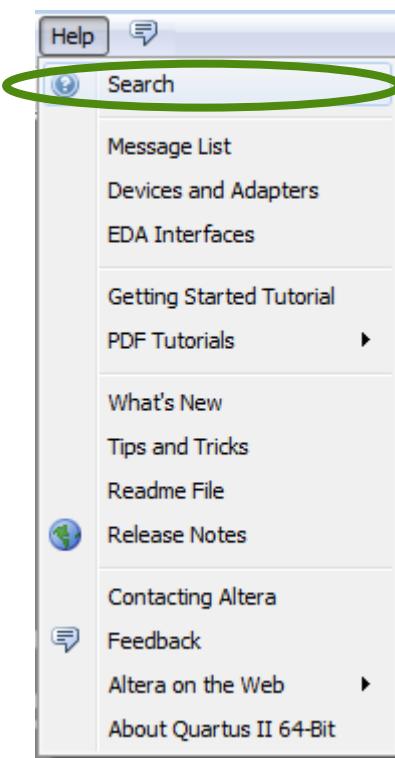
[Open Settings dialog box - Early Timing Estimate page](#)

Provides useful instructions on using the Quartus II software & links to settings.

Available sections include:

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

Built-In Help System



Quartus II Help v12.1 - Windows Internet Explorer

file:///C:/altera/12.1/quartus/common

Quartus II Help v12.1

Search Contents Index Forums Feedback

Shortcut to Altera forums

Quartus II Introduction

- What's New in Quartus II
- Using Quartus II Help
- Installation and Licensing
- Managing Projects
- Using Project Revisions
- Archiving Projects
- Exporting and Importing Versions
- Creating Designs
- Using Advisors for Design Optimizations
- Viewing Reports and Messages
- Using HDL with the Quartus II Software
- Using Altera Megafunctions
- Creating System-Level Designs
- Creating System-Level Designs
- Constraining Designs
- Compiling Designs

Welcome to the Quartus II Software

The Quartus II development software provides a complete design environment for system-on-a-programmable-chip (SOPC) design. Regardless of whether you use a personal computer or a Linux workstation, the Quartus II software ensures easy design entry, fast processing, and straightforward device programming. The following sections describe the general capabilities and design flows of the Quartus II software.

Show All

Expand all topics

Click any of the following flow icons for more information about that part of the design flow.

Design Entry

Includes block-based design, system-level design & software development

RTL Simulation

Internet | Protected Mode: Off

100%

Web browser-based allows for easy search in page

Interactive Tutorial

Quartus II Getting StartedTutorial - Windows Internet Explorer

C:\altera\12.1\quartus\common\help\tutorial\qtutorial.htm

Quartus II Getting StartedTutorial

Contents Home

QUARTUS® II TUTORIAL

Intro ✓

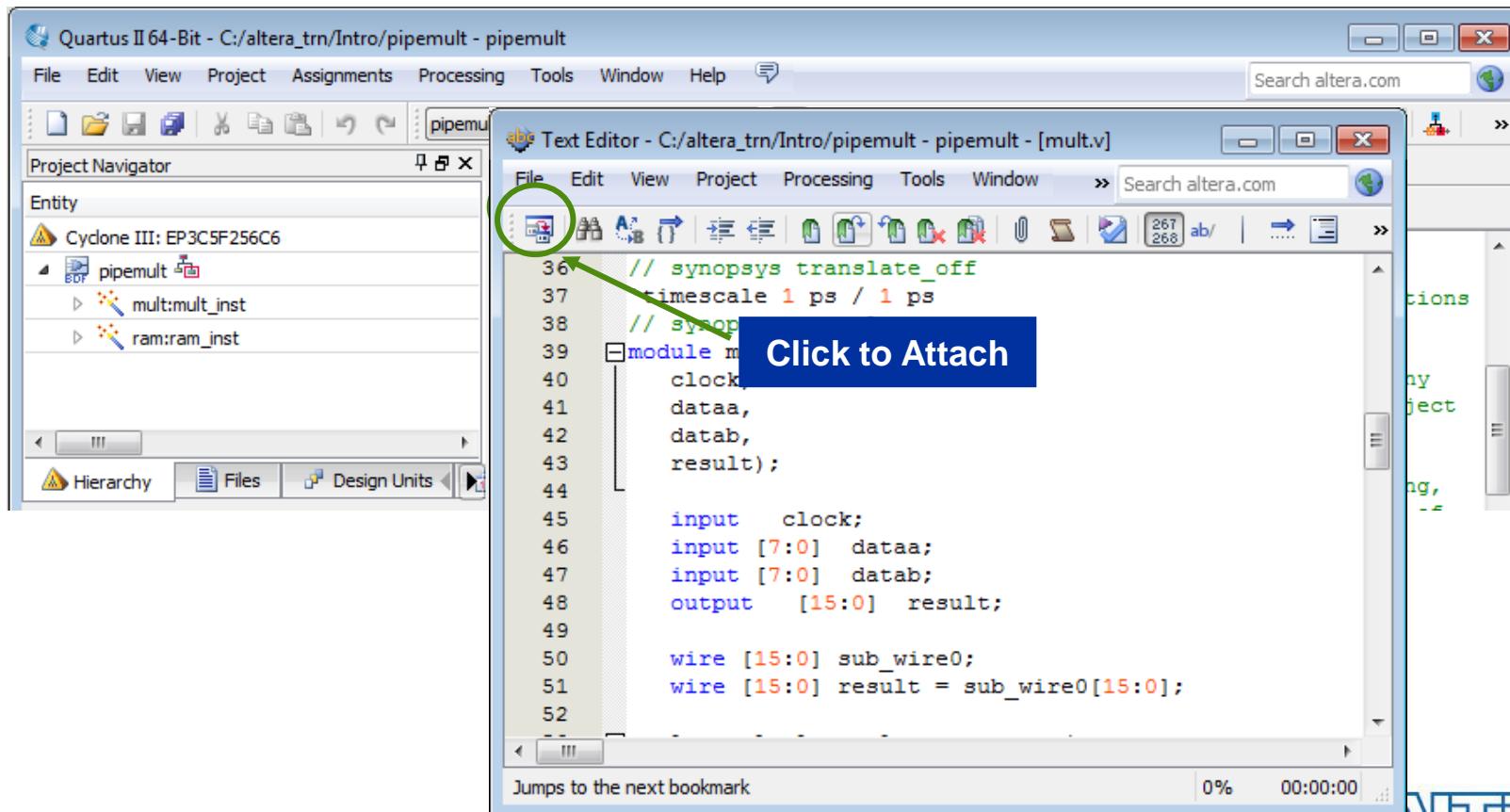
Quartus II Tutorial Modules

Module	Duration
Quartus II Introduction	(5 minutes)
Create a Design	(30 minutes)
Compile a Design	(40 minutes)
Run Timing Analysis	(40 minutes)
Configure a Device	(20 minutes)
Incremental Compilation	(40 minutes)
SignalTap II Logic Analyzer	(40 minutes)
Create a Qsys System	(20 minutes)

Help menu → Getting Started Tutorial or from the Getting Started window

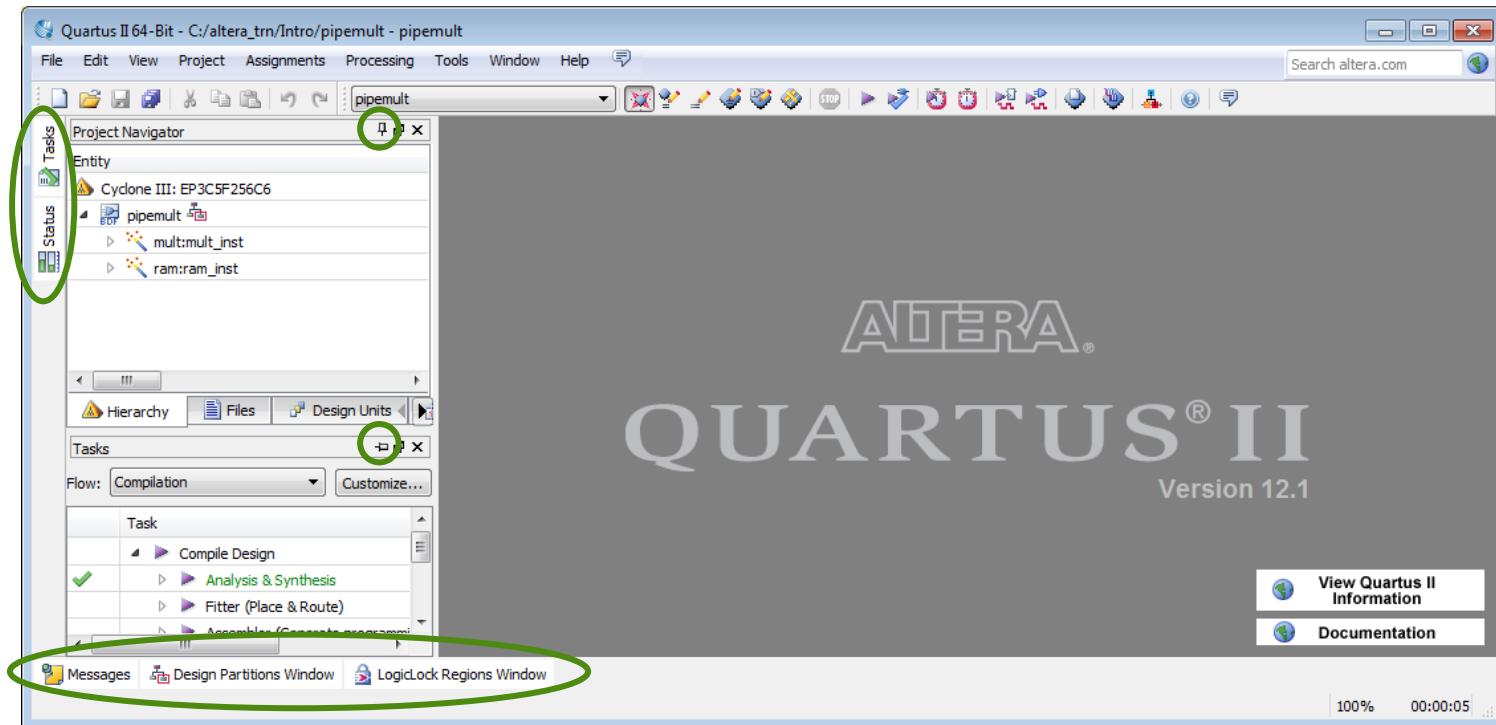
Detachable Windows

- Separate child windows from the Quartus II GUI frame (Window menu → Detach/Attach Window)



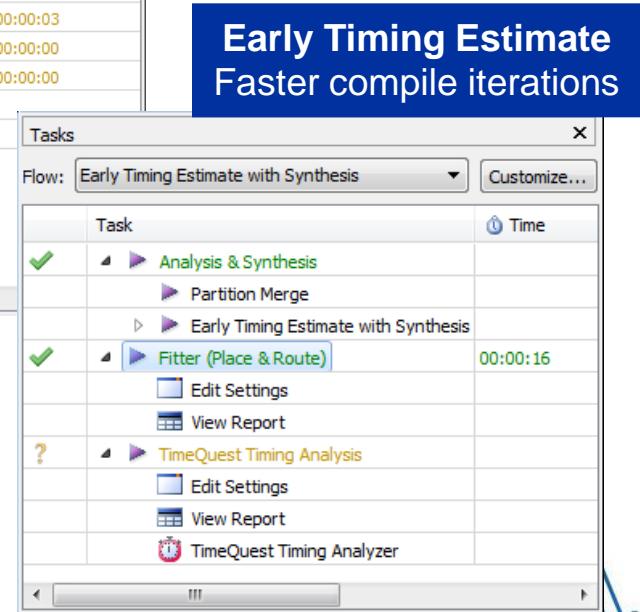
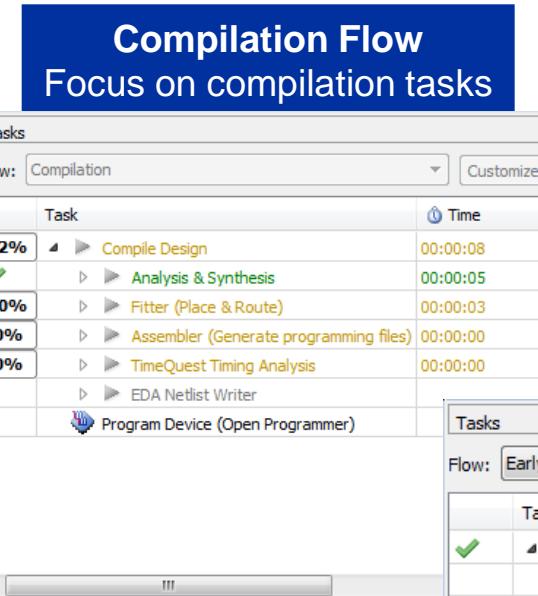
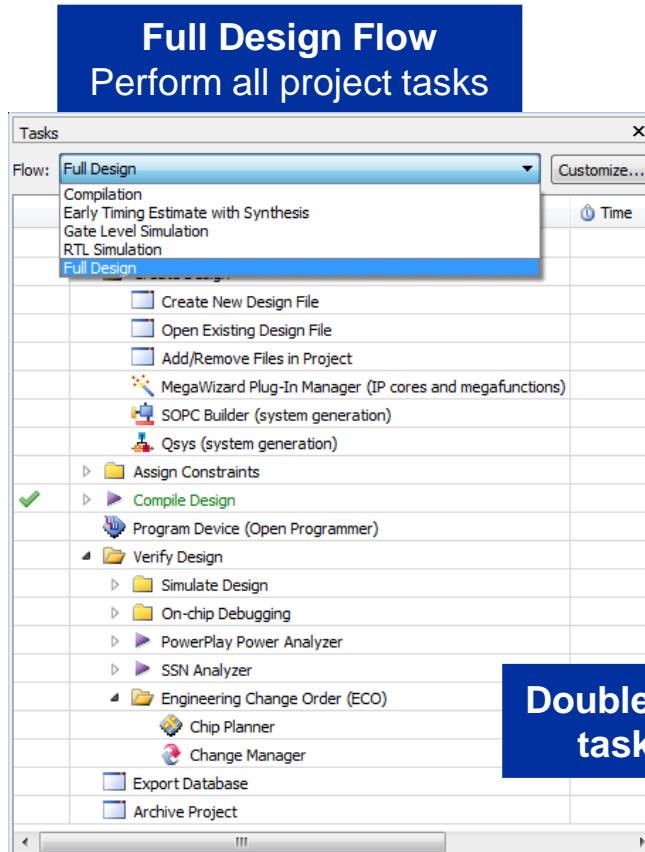
Auto Hide/Show Dockable Windows

- Allow user to hide/unhide dockable windows
- Mouse over the window bar and it will be visible
- User can pin the window to keep it visible



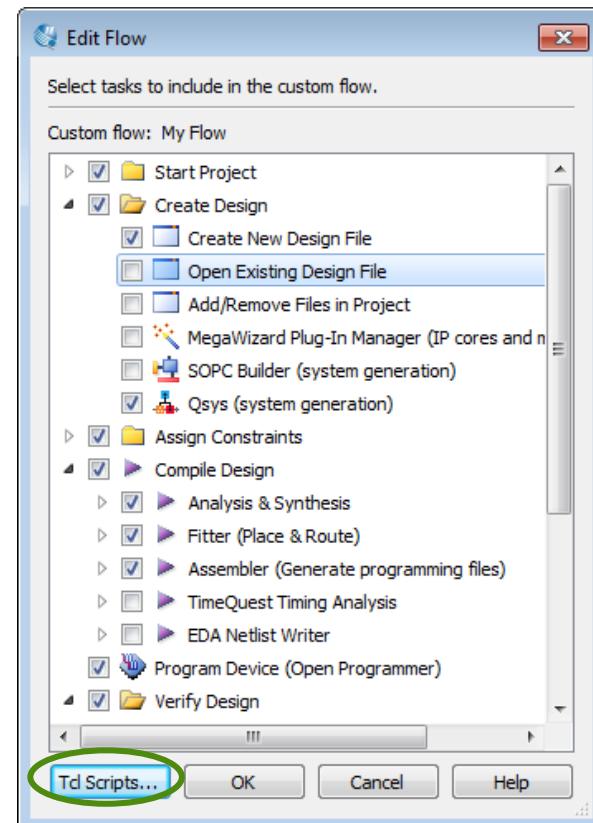
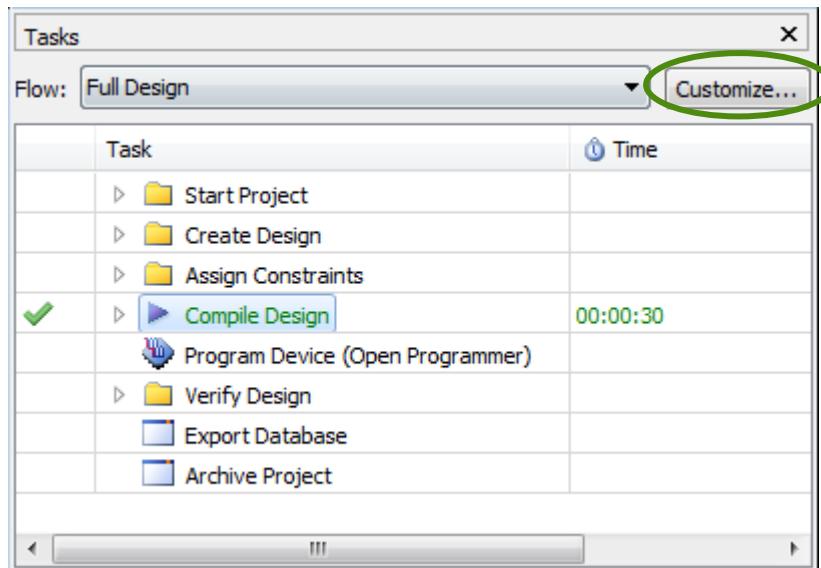
Tasks Window

- Easy access to most Quartus II functions
- Organized into related tasks within task flows



Custom Task Flow

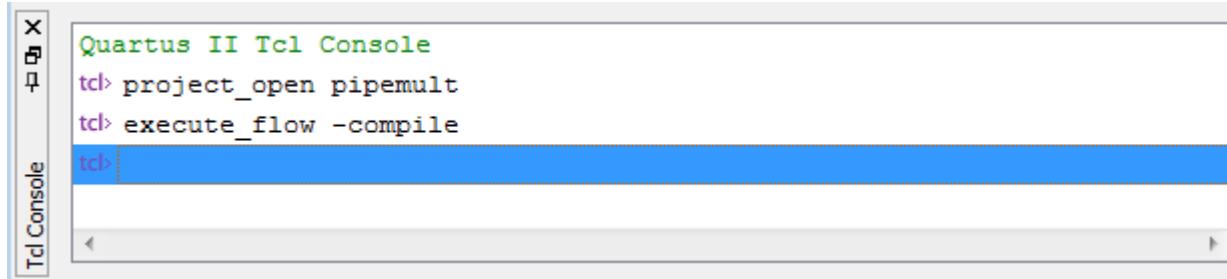
- Customize the Tasks display
- Add Tcl scripts for quick access



Tcl Console Window

- Enter and execute Tcl commands directly in the GUI

View menu → Utility Windows → Tcl Console



```
Quartus II Tcl Console
tcl> project_open pipemult
tcl> execute_flow -compile
tcl>
```

- Execute from command-line using Tcl shell
 - quartus_sh -shell
- Run complete scripts from Tools menu → Tcl Scripts...

Further Information

- **On-line training classes**
 - [Introduction to Tcl](#)
 - [Quartus II Software Tcl Scripting](#)
- **Quartus II Handbook Volume 2 Chapters 2 & 3**
- **Tcl references online**

Quartus II Software Design Series: Foundation

Quartus II Projects



© 2013 Altera Corporation—Public



Quartus II Projects Section Objectives

- **Create a project**
- **Open an existing project**
- **Name the basic Quartus II project files**
- **Archive a project**
- **Copy a project**
- **Create a project revision**

■ Description

- Collection of related design files & libraries
- Must have a designated top-level entity
- Target a single device
- Store settings in Quartus II Settings File (.QSF)
- Compiled netlist information stored in **db** folder in project directory

■ Create new projects with New Project Wizard

- Can be created using Tcl scripts

New Project Wizard

File menu

Quartus II 64-Bit - C:/altera_trn/Intro/pipemult -

File Edit View Project Assignments Processing Tools Help

New... Ctrl+N
Open... Ctrl+O
Close Ctrl+F4
New Project Wizard... Ctrl+J
Open Project... Save Project Close Project

Tasks

Flow: Full Design Customize...

Task Time

Start Project
Open New Project Wizard (highlighted)
Open Existing Project
Create Revision
Synchronize Project Libraries

Select working directory

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

New Project Wizard

Directory, Name, Top-Level Entity [page 1 of 5]

What is the working directory for this project?
C:\altera_trn\Intro

What is the name of this project?

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

Use Existing Project Settings...

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

Tcl: *project_new <project_name>*

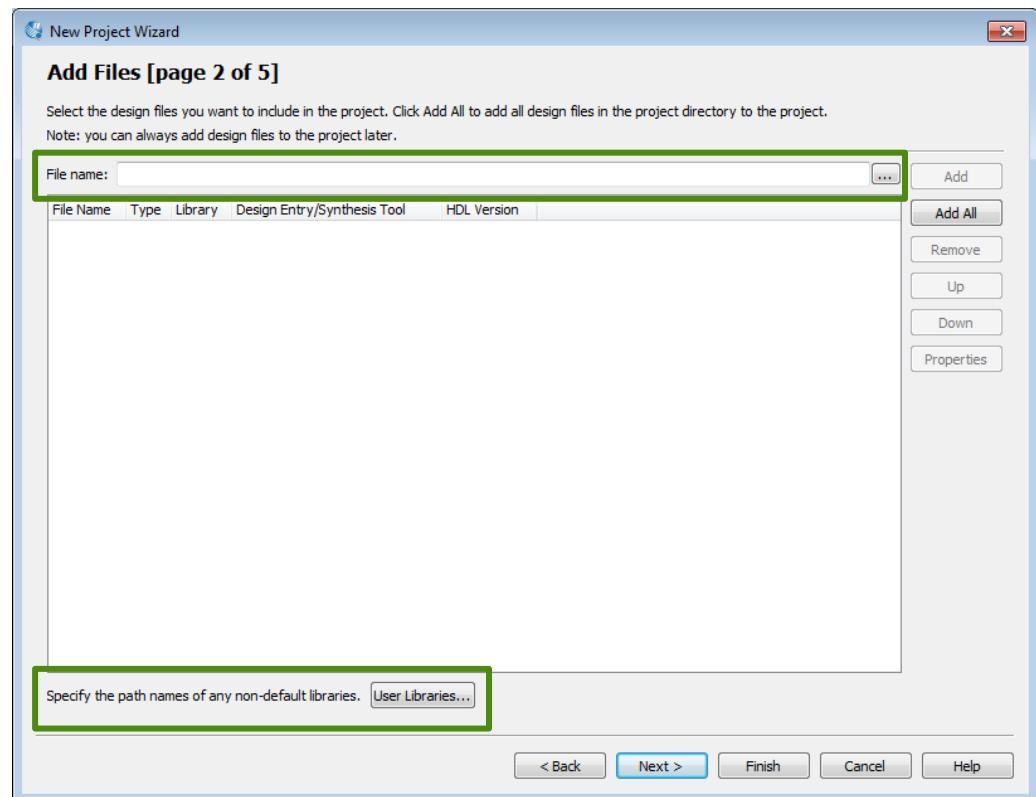
Add Files

■ Add design files

- Graphic
- VHDL
- Verilog
- SystemVerilog
- EDIF
- VQM

■ Add library paths

- User libraries
- MegaCore® library
- AMPP™ 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

New Project Wizard

Choose device family

Filter device list

Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.

Device family

Family: **Cyclone IV GX**

Devices: All

Target device

Auto device selected by the Filter

Specific device selected in 'Available devices' list

Other: n/a

Available devices:

Choose specific part from list

Name	Core Voltage	Package	Pin Count	Speed Grade	LAN Transmitter Channel PMA	LAN Receiver Channel PMA	GXB Receiver Channel PMA
EP4CGX15BF14A7	1.2V	14400	81	2			2
EP4CGX15BF14C6	1.2V	14400	81	2			2
EP4CGX15BF14C7	1.2V	14400	81	2			2
EP4CGX15BF14C8	1.2V	14400	81	2			2
EP4CGX15BF14I7	1.2V	14400	81	2			2
EP4CGX15BN11C7	1.2V	14400	81	2			2
EP4CGX15BN11C8	1.2V	14400	81	2			2

Companion device

HardCopy:

Limit DSP & RAM to HardCopy device resources

Show in 'Available devices' list

Package: Any

Pin count: Any

Speed grade: Any

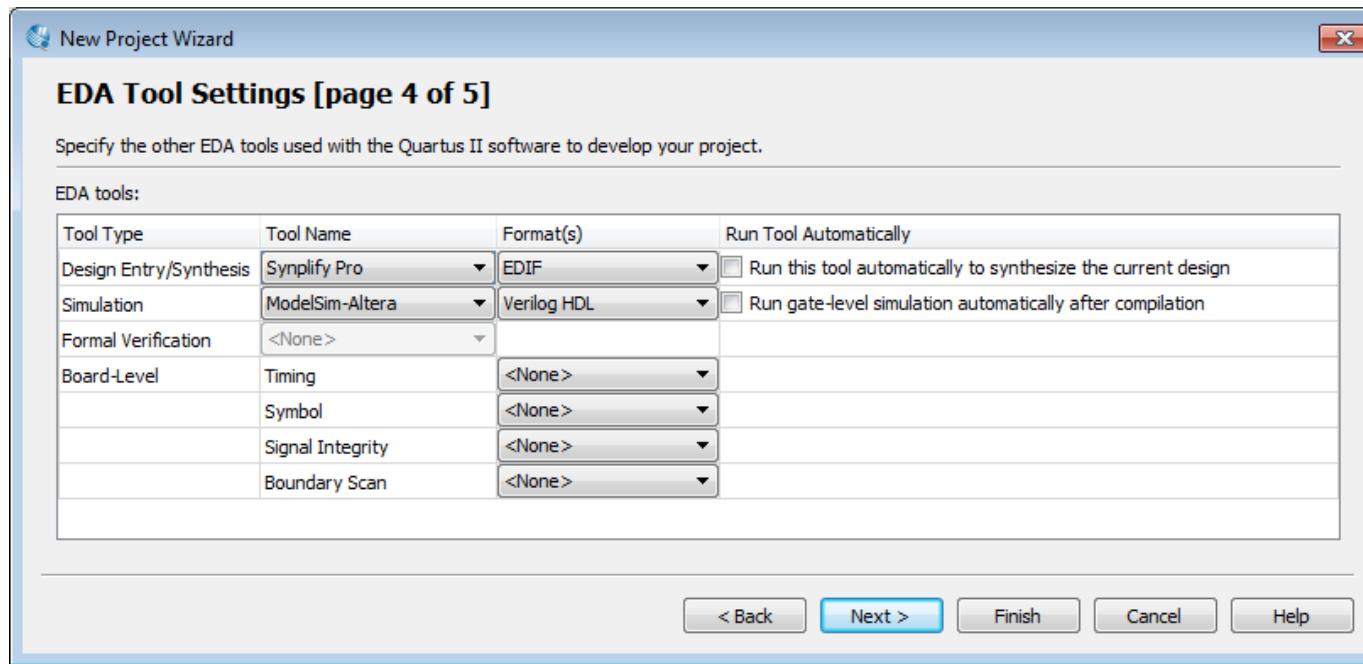
Name filter:

Show advanced devices HardCopy compatible only

Tcl: set_global_assignment -name FAMILY "device family name"
Tcl: set_global_assignment -name DEVICE <part_number>

EDA Tool Settings

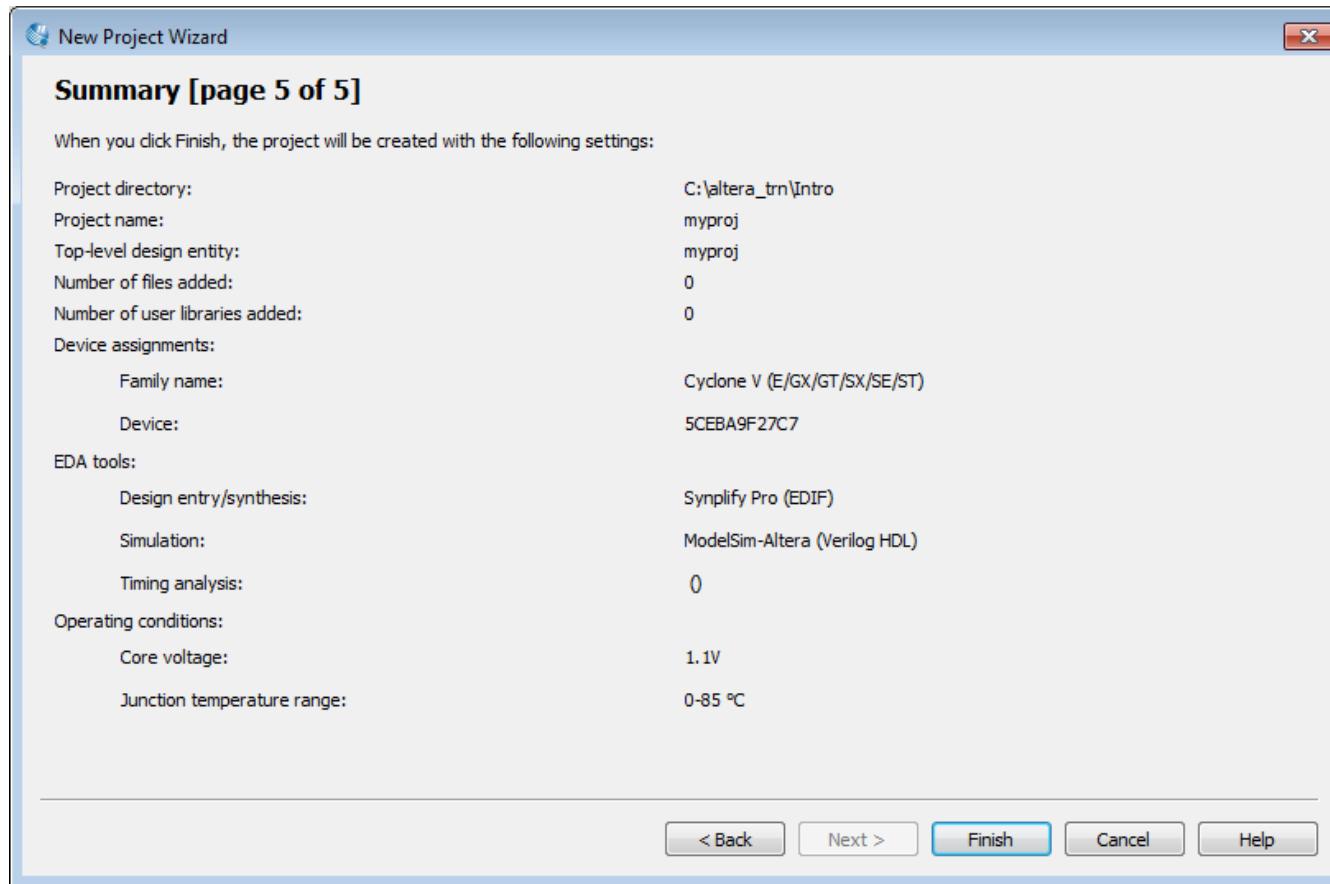
- Choose EDA tools and file formats
- Settings can be changed or added later



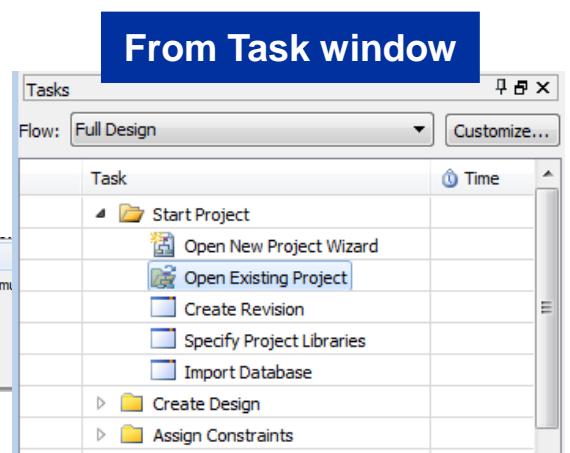
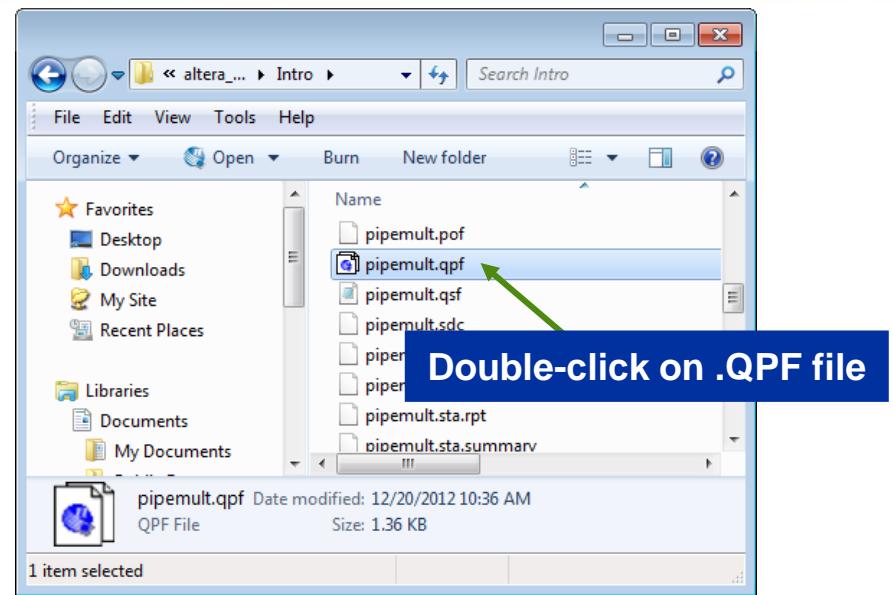
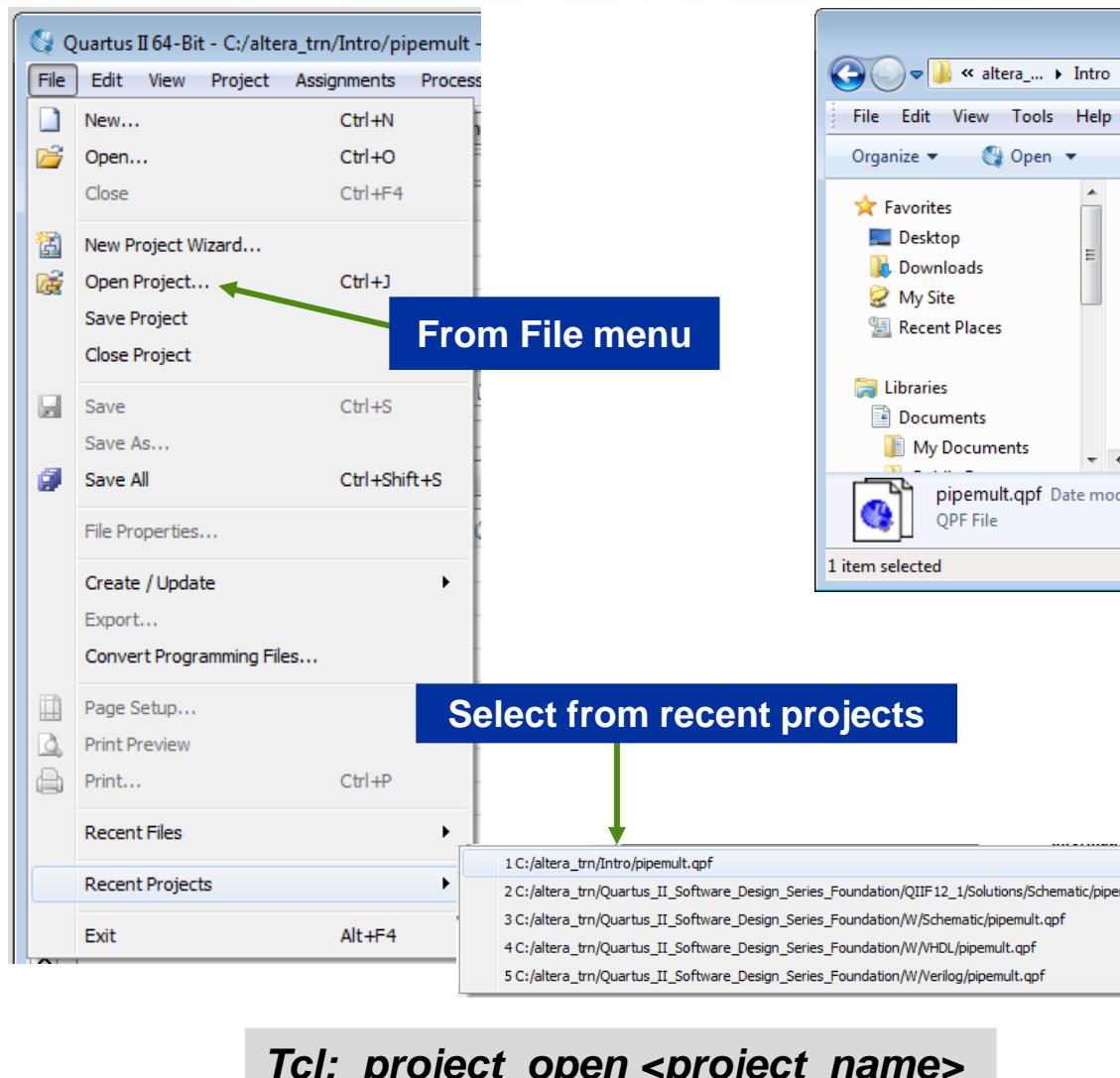
See *handbook for Tcl command format*

■ Review results & click Finish when done

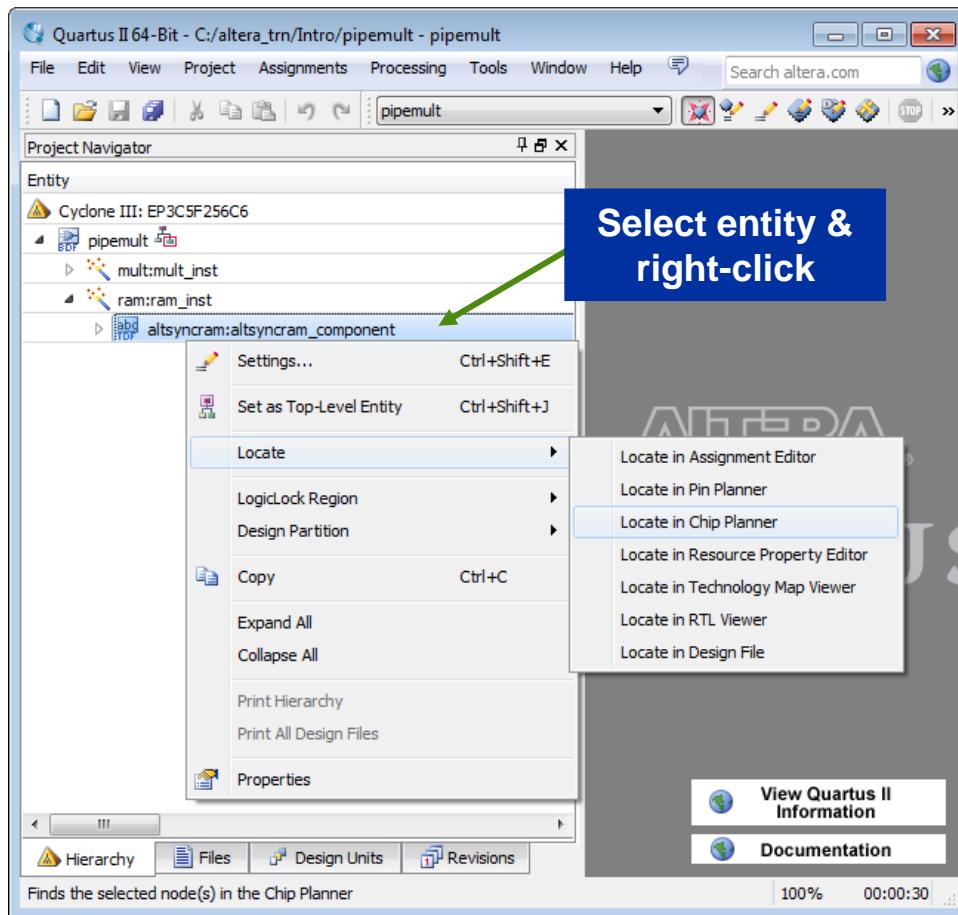
Review results & click Finish



Opening an Existing Project



Project Navigator – Hierarchy Tab



- Displays project hierarchy after project is analyzed

- **Uses**

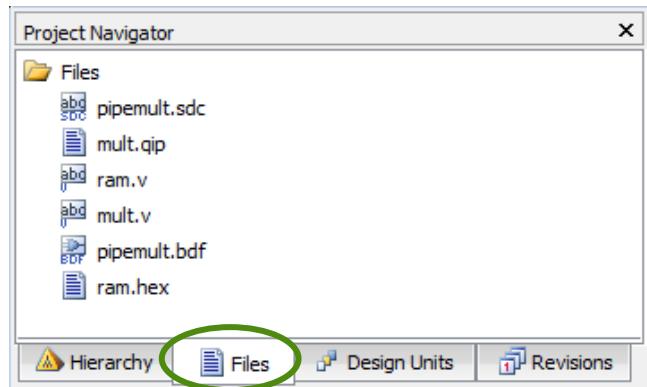
- Set top-level entity
- Set incremental design partition
- Make entity-level assignments
- Locate in design file or viewers/floorplans
- View resource usage

Full compilation or
Processing menu → Start → Start Hierarchy Elaboration

Files & Design Units Tabs

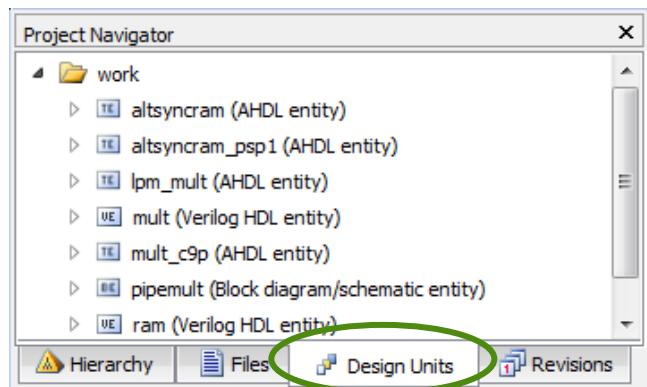
■ Files tab

- 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...**



■ Design Units tab

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



Quartus II Project Files & Folders

- **Quartus II Project File (.QPF)**
- **Quartus II Defaults File (.QDF)**
- **Quartus II Settings File (.QSF)**
- **db folder**
 - Contains compiled design information
 - May also see **incremental_db** for incremental compilation information
- **Synopsys Design Constraints (.SDC)**
 - Contains timing constraints

Project & Default Files

■ Quartus II Project File (QPF)

- Quartus II version
- Time stamp
- Active revision(s)

fir_filter.QPF

```
QUARTUS_VERSION = "12.1"
DATE = "08:37:10  Mar 13, 2012"

# Revisions

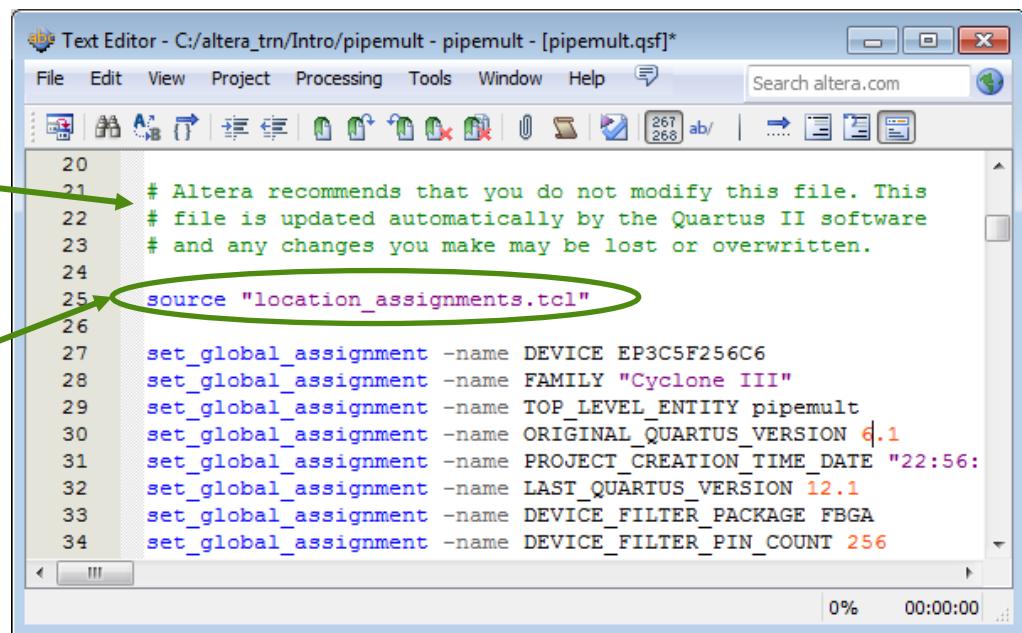
PROJECT_REVISION = "filtref"
PROJECT_REVISION = "filtref_new"
```

■ Quartus II Defaults Files (QDF)

- Stores Quartus II project setting & assignment defaults
- Example names: *assignment_defaults.qdf* or *<revision_name>_assignment_defaults.qdf*
- Found in local project or *altera\<version>\quartus\bin* directory
 - Copy stored in local project directory read before original version in *bin*

Quartus II Settings File (QSF)

- Stores all settings & assignments *except timing*
- Uses Tcl syntax
- Can be edited manually by user



The screenshot shows a Windows-style text editor window titled "Text Editor - C:/altera_trn/Intro/pipemult - pipemult - [pipemult.qsf]*". The file content is a Quartus II Settings File (QSF) with the following code:

```
20
21 # Altera recommends that you do not modify this file. This
22 # file is updated automatically by the Quartus II software
23 # and any changes you make may be lost or overwritten.
24
25 source "location_assignments.tcl"
26
27 set_global_assignment -name DEVICE EP3C5F256C6
28 set_global_assignment -name FAMILY "Cyclone III"
29 set_global_assignment -name TOP_LEVEL_ENTITY pipemult
30 set_global_assignment -name ORIGINAL_QUARTUS_VERSION 6.1
31 set_global_assignment -name PROJECT_CREATION_TIME_DATE "22:56:
32 set_global_assignment -name LAST_QUARTUS_VERSION 12.1
33 set_global_assignment -name DEVICE_FILTER_PACKAGE FBGA
34 set_global_assignment -name DEVICE_FILTER_PIN_COUNT 256
```

Two annotations are present:

- A blue box with the text "User comments start with #". An arrow points from this box to the first three lines of the file, which begin with the hash symbol "#".
- A blue box with the text "Source other Tcl/qsf files to organize assignments". An arrow points from this box to the line "source \"location_assignments.tcl\"". This line is highlighted with a green oval.

See "Quartus II Settings File Reference Manual" for more details on QSF assignments & syntax

Constraint Files & Assignment Priority

1. QSF

- Highest priority
- Assignments always used from here first

2. 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 Quartus II software

3. QDF located in project directory

- *assignment_defaults.qdf*
- Created automatically in project directory when project archived & restored

4. QDF located in Quartus II *bin* directory

- Lowest priority
- Assignments only used if not found in higher priority files

- Project archive & restore
- Project copy
- Revisions

■ Creates 2 files

- Compressed Quartus II 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)

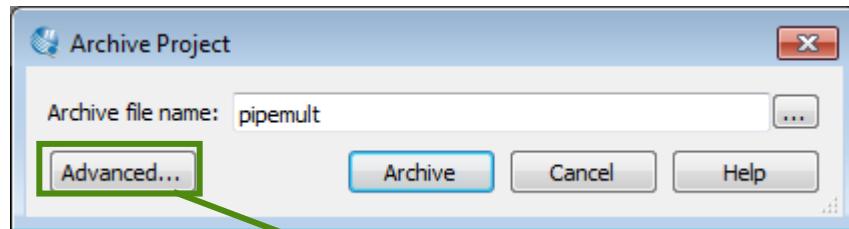
■ Example uses

- File storage (version control)
- Project handoff - Useful for sending to Altera support

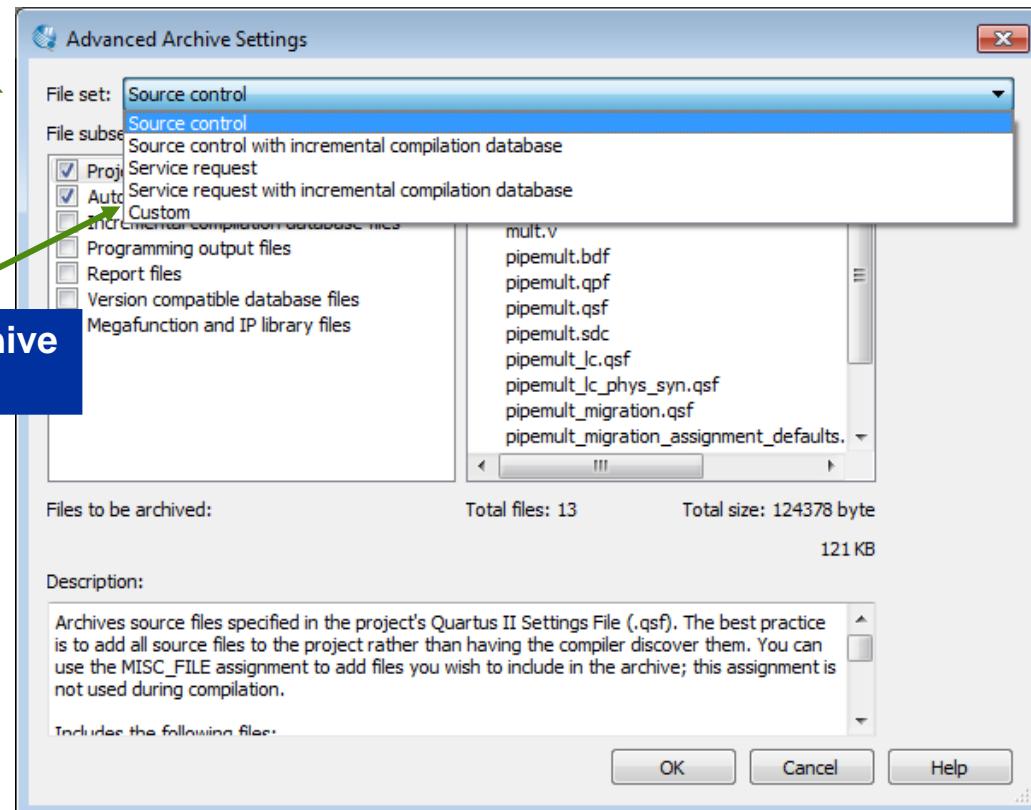
■ Design files referenced from user libraries are included in archive

Tcl: project_archive <project_name>

Project Archive (cont.)



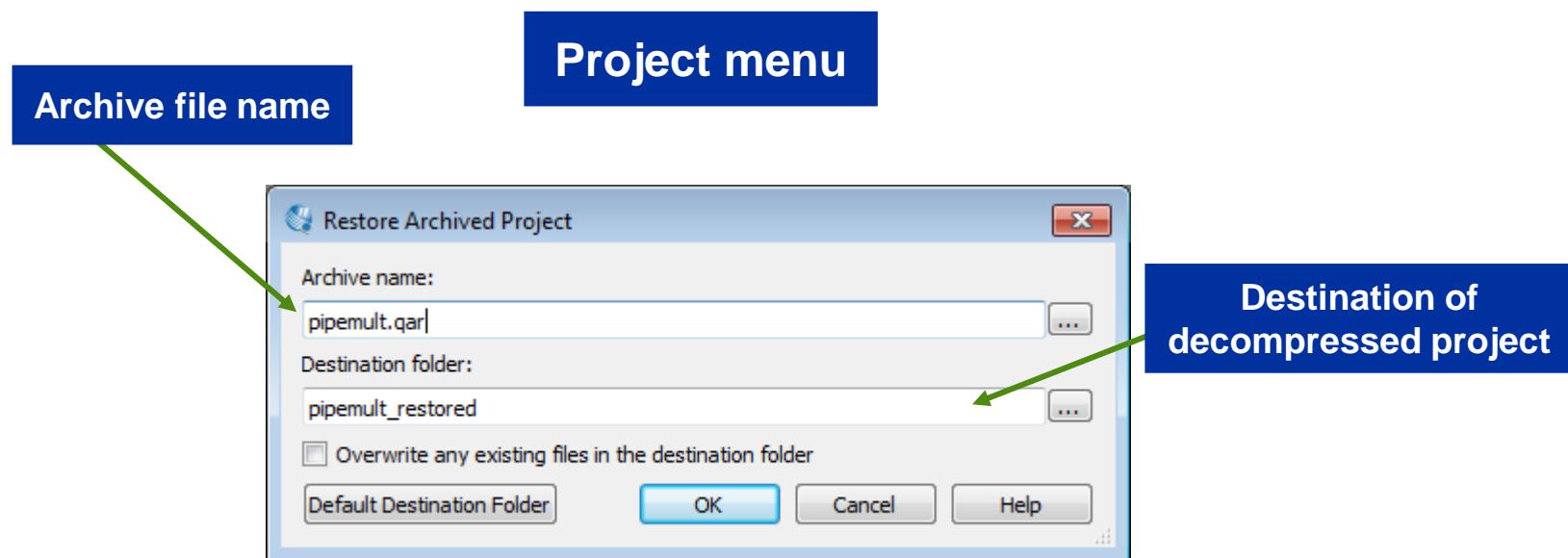
Project menu or
Tasks window



- Preset files included in archive
- Create custom archive

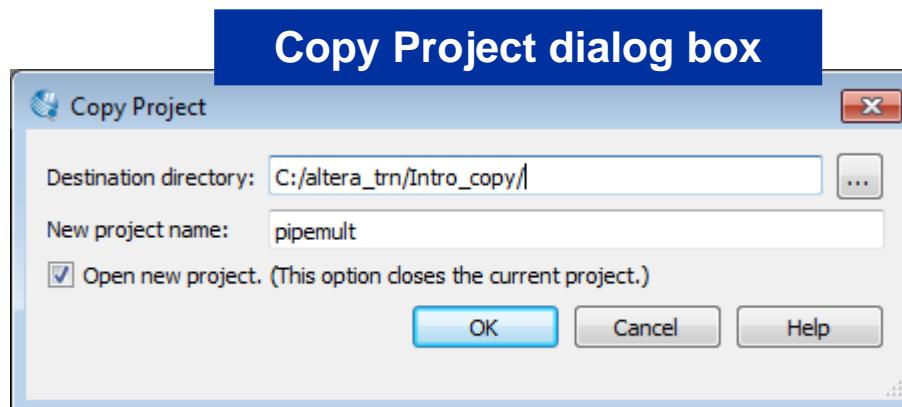
Restore Archived Project

- Decompresses .QAR into specified directory



Tcl: project_restore <archive_file>

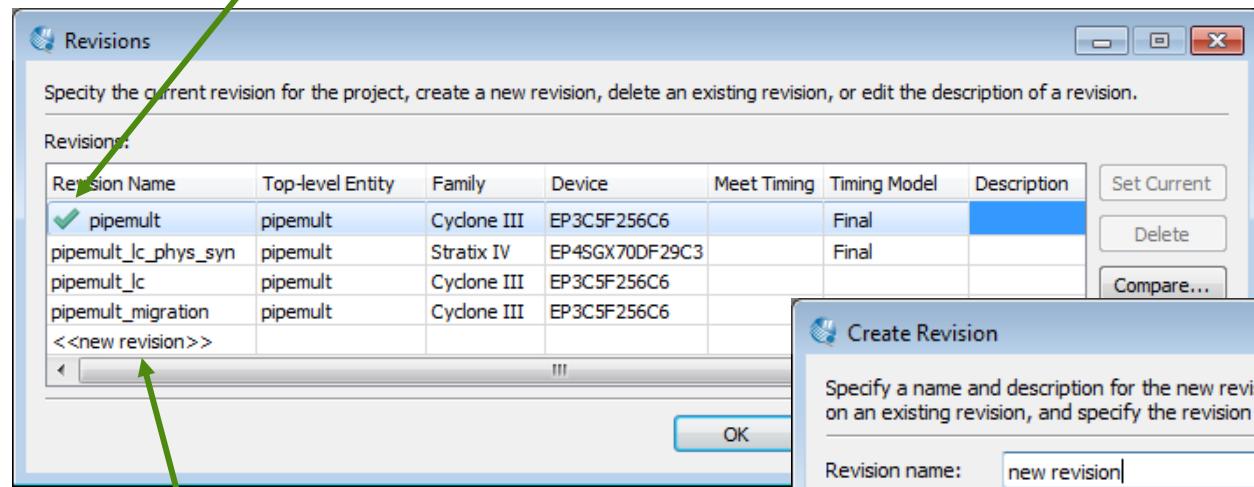
- **Copy & save *exact* duplicate of project in new directory**
 - Project file (.QPF)
 - Design files
 - Settings files
- **Example use - Duplicating work before editing design files**
- **User libraries are *not* copied; check paths**
- **New local QDF not created; only copies QDF if it exists**



Revisions

- **Explore new sets of constraints**
- **Compile options without losing previous work**
- **Compare results between revisions**
- **Revision-specific project files generated and stored in db directory**
 - Copy and update current revision files (no recompile required)
 - Generate new ones when new revision created and compiled

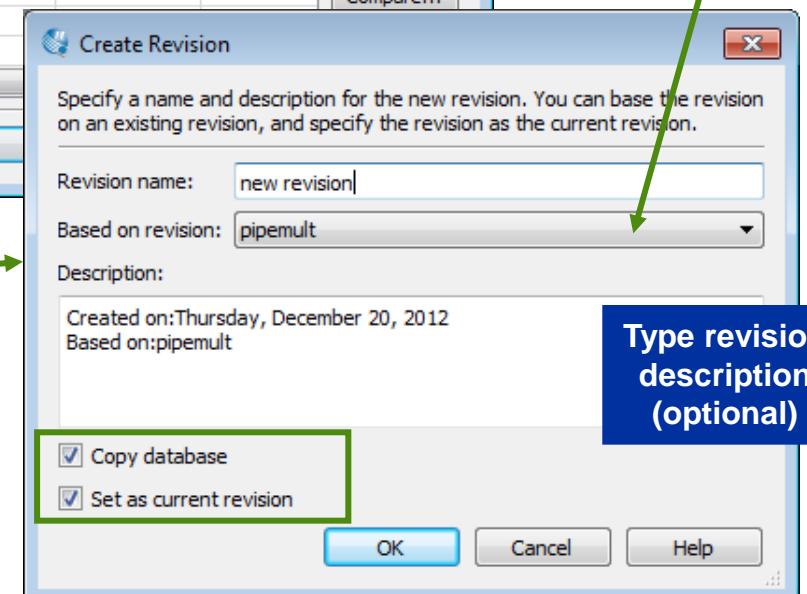
Creating a Revision



Project menu or
Tasks window

Base revision on any
previous revision

Create new revision

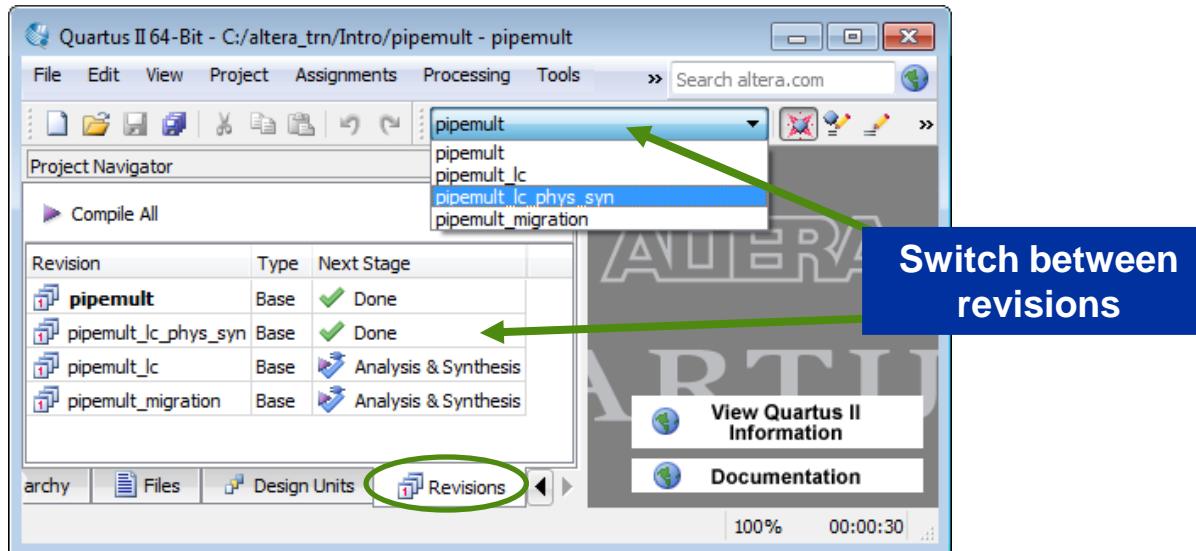


Type revision
description
(optional)

Tcl: *create_revision <revision_name>*

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)
 - *<revision_name>_description.TXT*



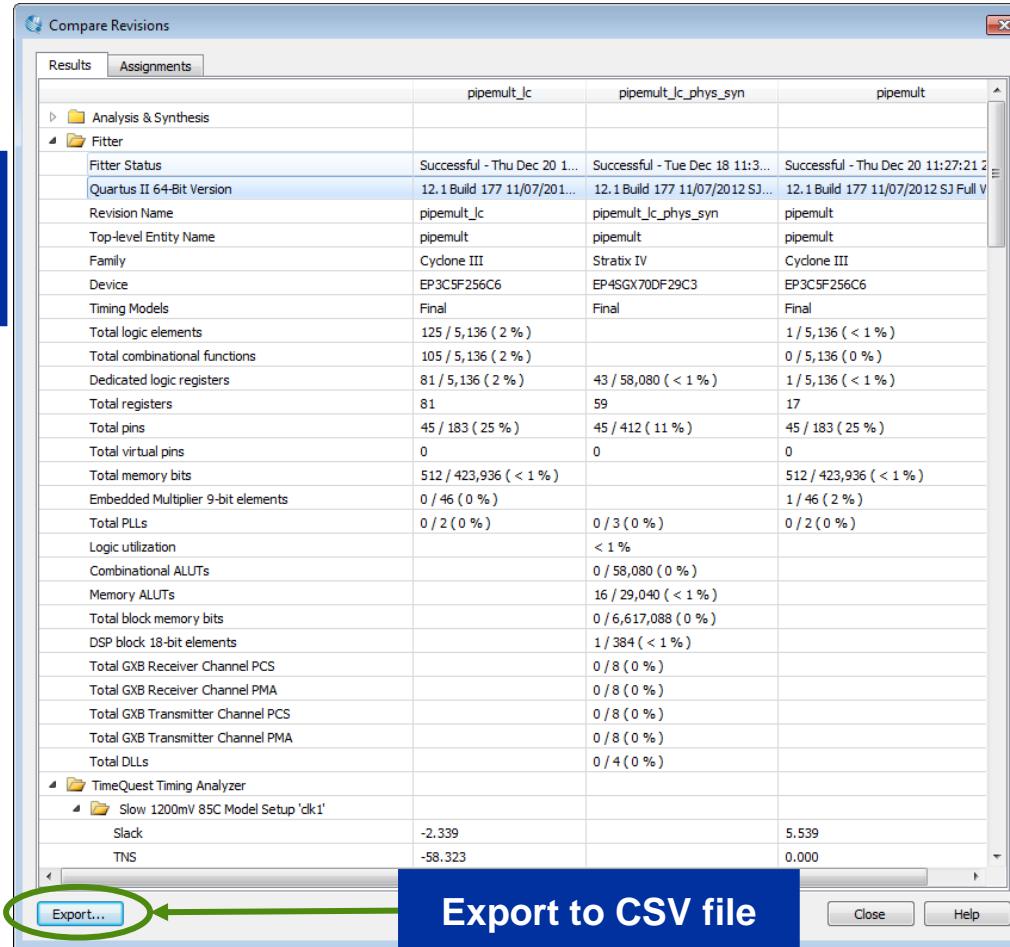
Tcl: project_open -revision <revision_name> <project_name>

Tcl: set_current_revision <revision_name>

Compare Revisions

■ Click Compare... from the Revisions dialog box

Detailed summary of revision assignments and results



The screenshot shows the 'Compare Revisions' dialog box with the 'Results' tab selected. The dialog box displays a detailed comparison of three projects: pipemult_lc, pipemult_lc_phys_syn, and pipemult. The comparison includes sections for Analysis & Synthesis, Fitter, and TimeQuest Timing Analyzer. The Fitter section provides a breakdown of logic elements, registers, pins, and memory bits. The TimeQuest Timing Analyzer section shows slack and TNS values for a specific model setup. At the bottom left, a green arrow points to the 'Export...' button, which is highlighted with a blue box and labeled 'Export to CSV file'.

	pipemult_lc	pipemult_lc_phys_syn	pipemult
Analysis & Synthesis			
Fitter			
Fitter Status	Successful - Thu Dec 20 1...	Successful - Tue Dec 18 11:3...	Successful - Thu Dec 20 11:27:21 2...
Quartus II 64-Bit Version	12.1 Build 177 11/07/201...	12.1 Build 177 11/07/2012 SJ...	12.1 Build 177 11/07/2012 SJ Full V...
Revision Name	pipemult_lc	pipemult_lc_phys_syn	pipemult
Top-level Entity Name	pipemult	pipemult	pipemult
Family	Cyclone III	Stratix IV	Cyclone III
Device	EP3C5F256C6	EP4SGX70DF29C3	EP3C5F256C6
Timing Models	Final	Final	Final
Total logic elements	125 / 5,136 (2 %)		1 / 5,136 (< 1 %)
Total combinational functions	105 / 5,136 (2 %)		0 / 5,136 (0 %)
Dedicated logic registers	81 / 5,136 (2 %)	43 / 58,080 (< 1 %)	1 / 5,136 (< 1 %)
Total registers	81	59	17
Total pins	45 / 183 (25 %)	45 / 412 (11 %)	45 / 183 (25 %)
Total virtual pins	0	0	0
Total memory bits	512 / 423,936 (< 1 %)		512 / 423,936 (< 1 %)
Embedded Multiplier 9-bit elements	0 / 46 (0 %)		1 / 46 (2 %)
Total PLLs	0 / 2 (0 %)	0 / 3 (0 %)	0 / 2 (0 %)
Logic utilization		< 1 %	
Combinational ALUTs		0 / 58,080 (0 %)	
Memory ALUTs		16 / 29,040 (< 1 %)	
Total block memory bits		0 / 6,617,088 (0 %)	
DSP block 18-bit elements		1 / 384 (< 1 %)	
Total GXB Receiver Channel PCS		0 / 8 (0 %)	
Total GXB Receiver Channel PMA		0 / 8 (0 %)	
Total GXB Transmitter Channel PCS		0 / 8 (0 %)	
Total GXB Transmitter Channel PMA		0 / 8 (0 %)	
Total DLLs		0 / 4 (0 %)	
TimeQuest Timing Analyzer			
Slow 1200mV 85C Model Setup 'clk1'			
Slack	-2.339		5.539
TNS	-58.323		0.000

Test Your Knowledge: Quartus II Projects

1. What are the (4) basic Quartus II project files?

A.

2. Which of the basic project files is unique for each revision?

A.

Exercise 1 Demonstration

*Demo should open automatically click
the link above if it doesn't*

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, and revisions provide easy-to-use project management

Project Support Resources

- ***Managing Quartus II Projects*** chapter in Volume 2 of the Quartus II Handbook

Quartus II Handbook always available online in pdf format at (*Literature* link from home page):

<http://www.altera.com/literature/lit-qts.jsp>

Quartus II Software Design Series: Foundation

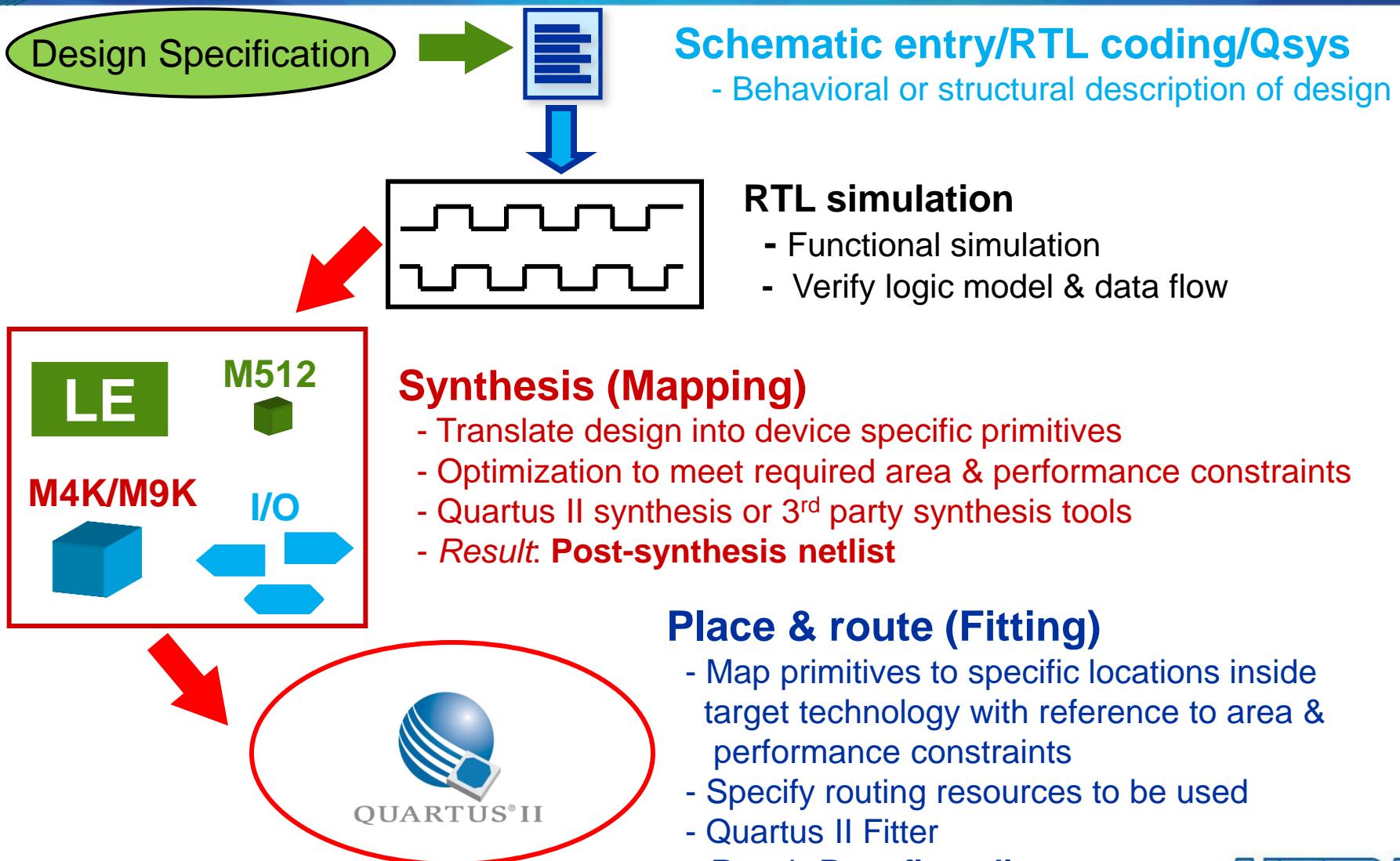
Design Methodology



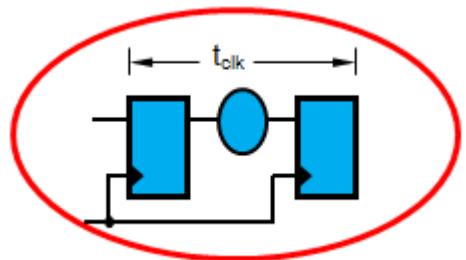
© 2013 Altera Corporation—Public



Typical PLD Design Flow

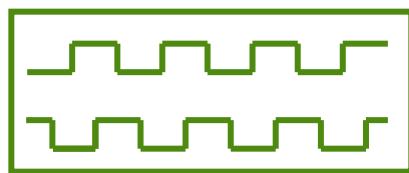


Typical PLD Design Flow



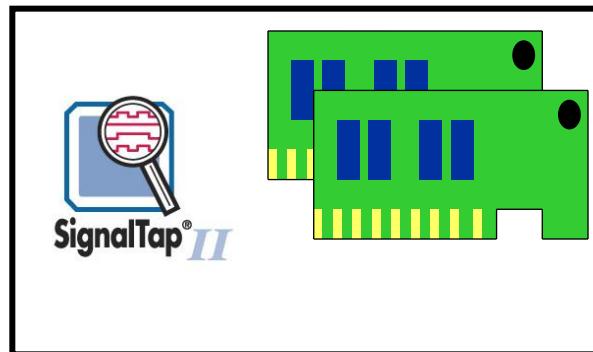
Timing analysis (TimeQuest Timing Analyzer)

- Verify performance specifications were met
- Static timing analysis



Gate level simulation (optional)

- Simulation with timing delays taken into account
- Verify design will work in target technology



PC board simulation & test

- Simulate board design
- Program & test device on board
- Use **SignalTap™ II** Logic Analyzer or other on-chip tools for debugging

Quartus II Software Design Series: Foundation

Design Entry



© 2013 Altera Corporation—Public



Design Entry Methods

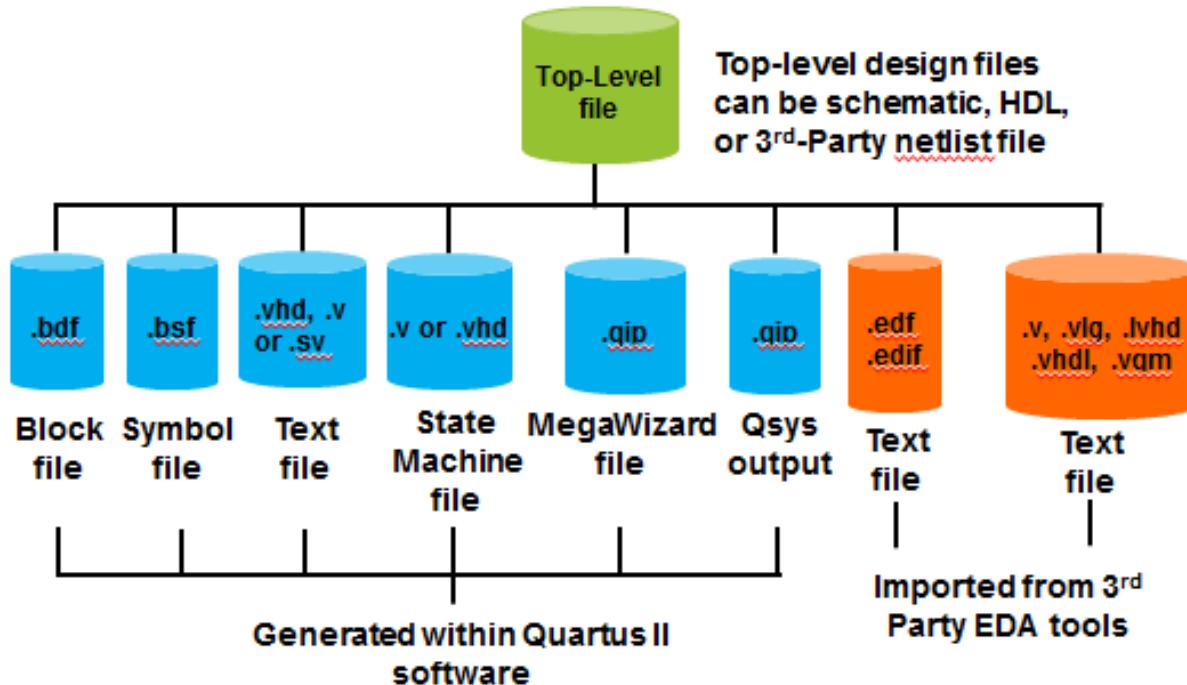
■ Quartus II design entry

- Text editor
 - VHDL
 - Verilog or SystemVerilog
- Schematic editor
 - Block Diagram File
- System editor
 - Qsys
- State machine editor
 - HDL from state machine file
- Memory editor
 - HEX
 - MIF

■ 3rd-party EDA tools

- EDIF 2 0 0
- Verilog Quartus Mapping (.VQM)

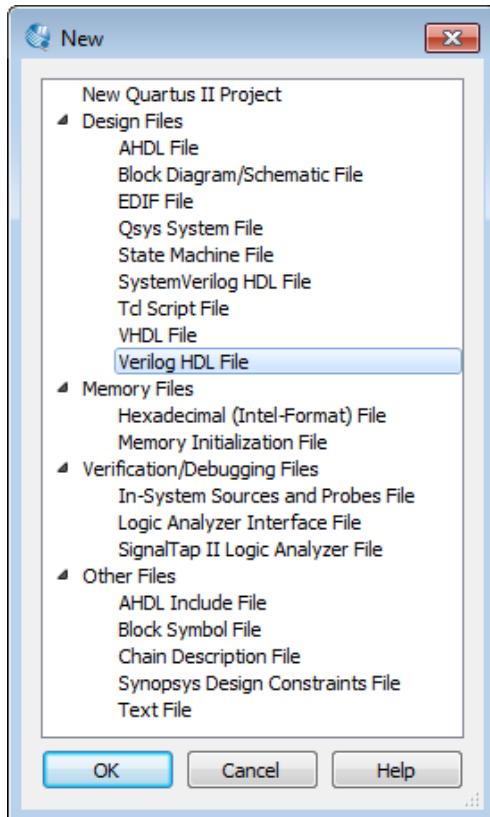
Design Entry File Types Supported



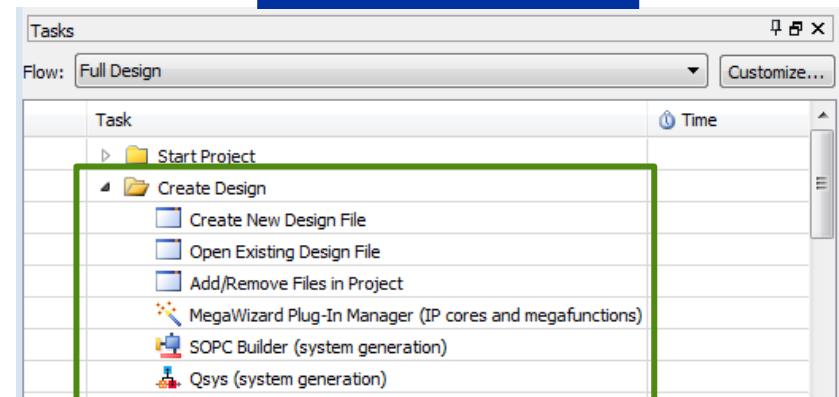
- Mixing & matching design files allowed

Creating New Design Files (& Others)

File → New or  in Toolbar



Tasks window



■ Quartus II 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 TimeQuest timing analyzer
- Preview/editing of full design and construct HDL templates

■ Enter text description

- VHDL (.vhd, .vhdl)
- Verilog (.v, .vlg, .Verilog, .vh)
- SystemVerilog (.sv)

- **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 Quartus II integrated synthesis to synthesize**
- **View supported commands in built-in help**

*Learn more about HDL in Altera HDL
customer training classes*

Text Editor Features

The screenshot shows the Altera Text Editor interface with several features highlighted:

- Find/highlight matching delimiters**: A callout box with a green arrow points to the magnifying glass icon in the toolbar, which is used for finding and highlighting matching delimiters.
- Bookmarks (on/off/jump to)**: A callout box with a green arrow points to the bookmark icon in the toolbar, which is used for managing and jumping to bookmarks.
- Insert Template (Edit menu)**: A callout box with a green arrow points to the "Edit" menu, which contains the "Insert Template" option.
- Preview window: edit before inserting & save as user template**: A callout box with a green arrow points to the "Insert Template" dialog box, which shows a preview of the template code and options to save or insert.
- Collapse/expand functions**: A callout box with a green arrow points to the collapse/expand icon in the toolbar, which is used for collapsing or expanding code blocks.

Text Editor Features

File Edit View Project Processing Tools Window Help

Search altera.com

39 module ram (

40 clock,

41 data,

42 rdaddress,

43 wraddress,

44 wren,

45 q);

46

47 input clock;

48 input [15:0] data;

49 input [4:0] rdaddress;

50 input [4:0] wraddress;

51 input wren;

52 output [15:0] q;

53

54 wire [15:0] sub_wire0;

55 wire [15:0] q = sub_wire0[15:0];

56

57 altsyncram altsyncram_component (

58 .wren_a (wren),

59 .clock0 (clock),

60 .address_a (wraddress),

61 .address_b (rdaddress),

62 .data_a (data),

63);

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

Quartus II Text Editor Features

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

```
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
    rx_inclock
    rx_locked
    rx_out
    common_rx_tx_pll => "OFF",
    data_align_rollover => 4,
    data_rate => "UNUSED",
    deserialization_factor => 4,
    dpa_initial_phase_value => 0,
```

```
input  clock_ena;
input  signed [WIDTH-1:0] a0;
input  signed [WIDTH-1:0] b0;
input  signed [WIDTH-1:0] a1;
input  signed [WIDTH-1:0] b1;
output signed [2*WIDTH-1:0] rout;
output signed [2*WIDTH-1:0] iout;

reg  signed [WIDTH-1:0] a0_reg;
reg  signed [WIDTH-1:0] b0_reg;
reg  signed [WIDTH-1:0] a1_reg;
reg  signed [WIDTH-1:0] b1_reg;
reg  signed [WIDTH-1:0] a2_reg;
reg  signed [WIDTH-1:0] b2_reg;
reg  signed [WIDTH-1:0] a3_reg;
reg  signed [WIDTH-1:0] b3_reg;

reg  signed [2*WIDTH-1:0] rout;
reg  signed [2*WIDTH-1:0] iout;
```

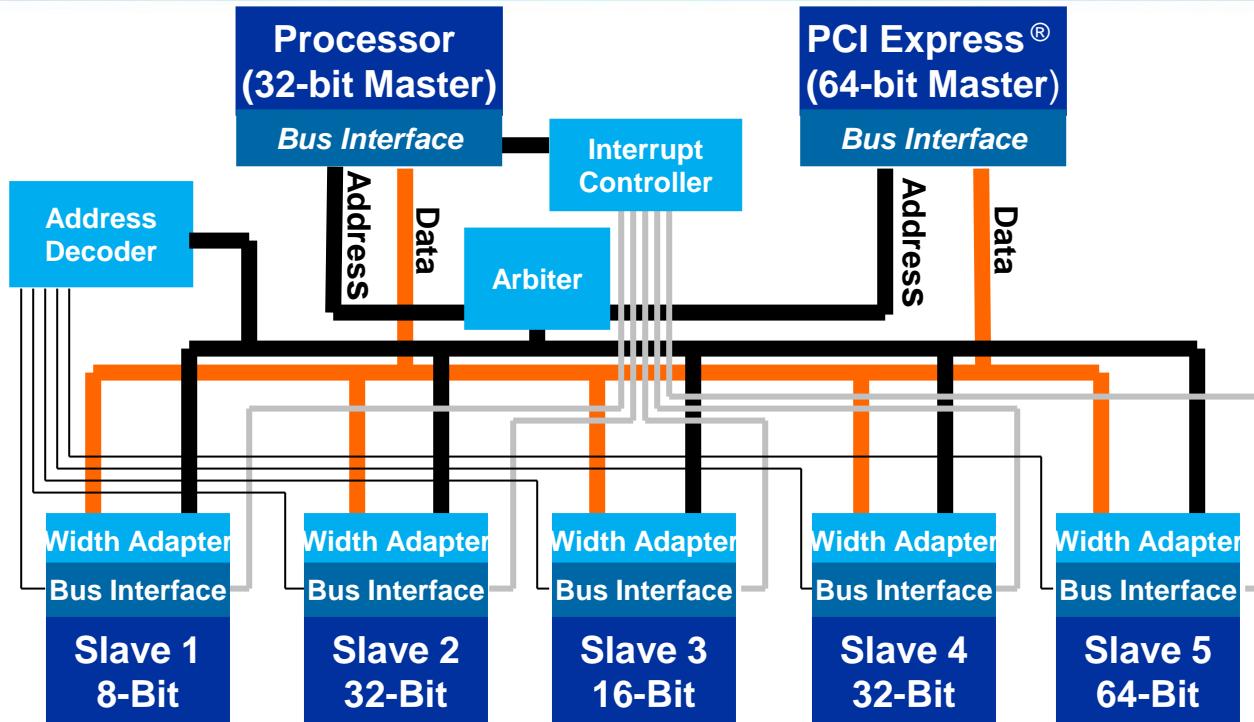
```
1 //18 bit complex input, 36 bit real and imaginary outputs
2 // (a0+j*b0)*(a1+j*b1) = (a0*a1-b0*b1)+j*(a0*a1+b0*b1)
3 module half_DSP_block (clock, areset, clock_ena, a0, b0, a1, b1, rout, iout)
4   parameter WIDTH = 18;
5   int Error(10161); Verilog HDL error at half_DSP_block.v(3); object "clock"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

- **Full-featured schematic design capability**
- **Schematic Editor uses**
 - Create simple test designs to understand the functionality of an Altera megafunction - PLL, LVDS I/O, memory, etc...
 - Create top-level schematic for easy viewing & connection
 - Conversion utilities

Note: **Schematic entry online training available:** [Using the Quartus II Software: Schematic Design](#)

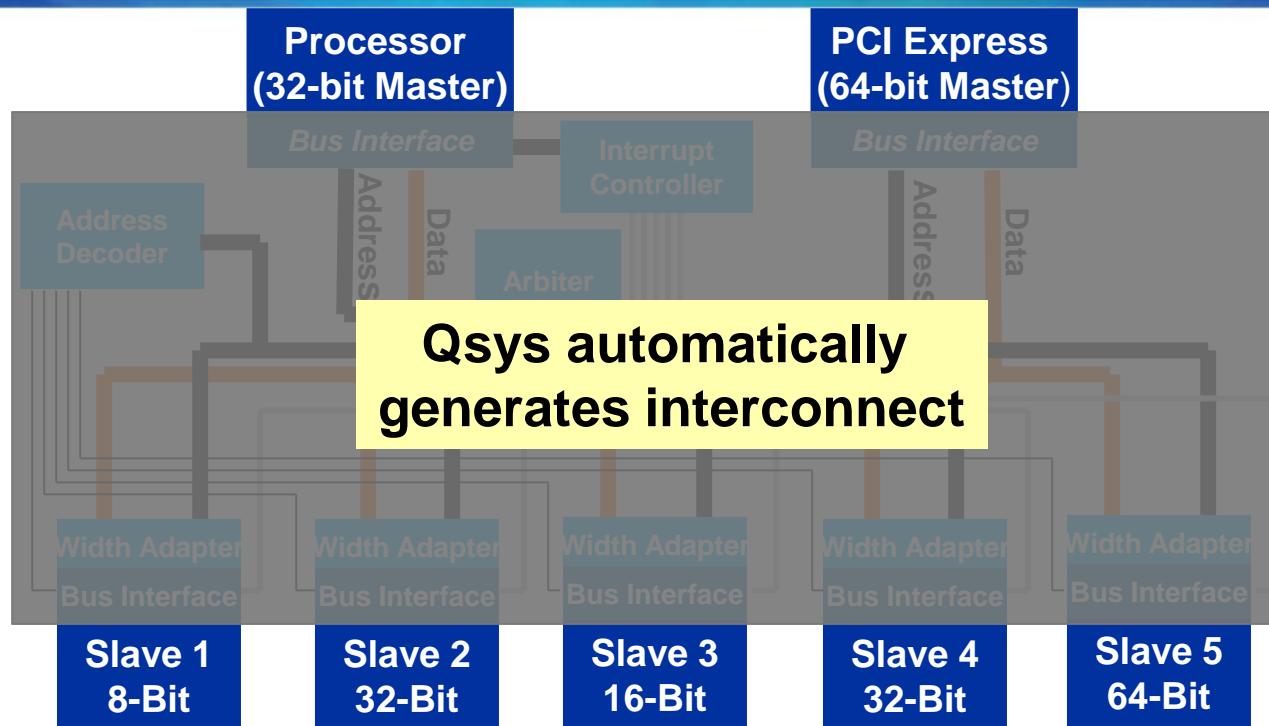
- **Simplifies complex system development**
 - Automatic interconnect generation
- **Raises the level of design abstraction**
 - High level design and system visualization
- **Provides a standard platform: IP integration, custom IP authoring, IP verification**
- **Enables design re-use**
- **Scales easily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

Traditional System Design



- 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

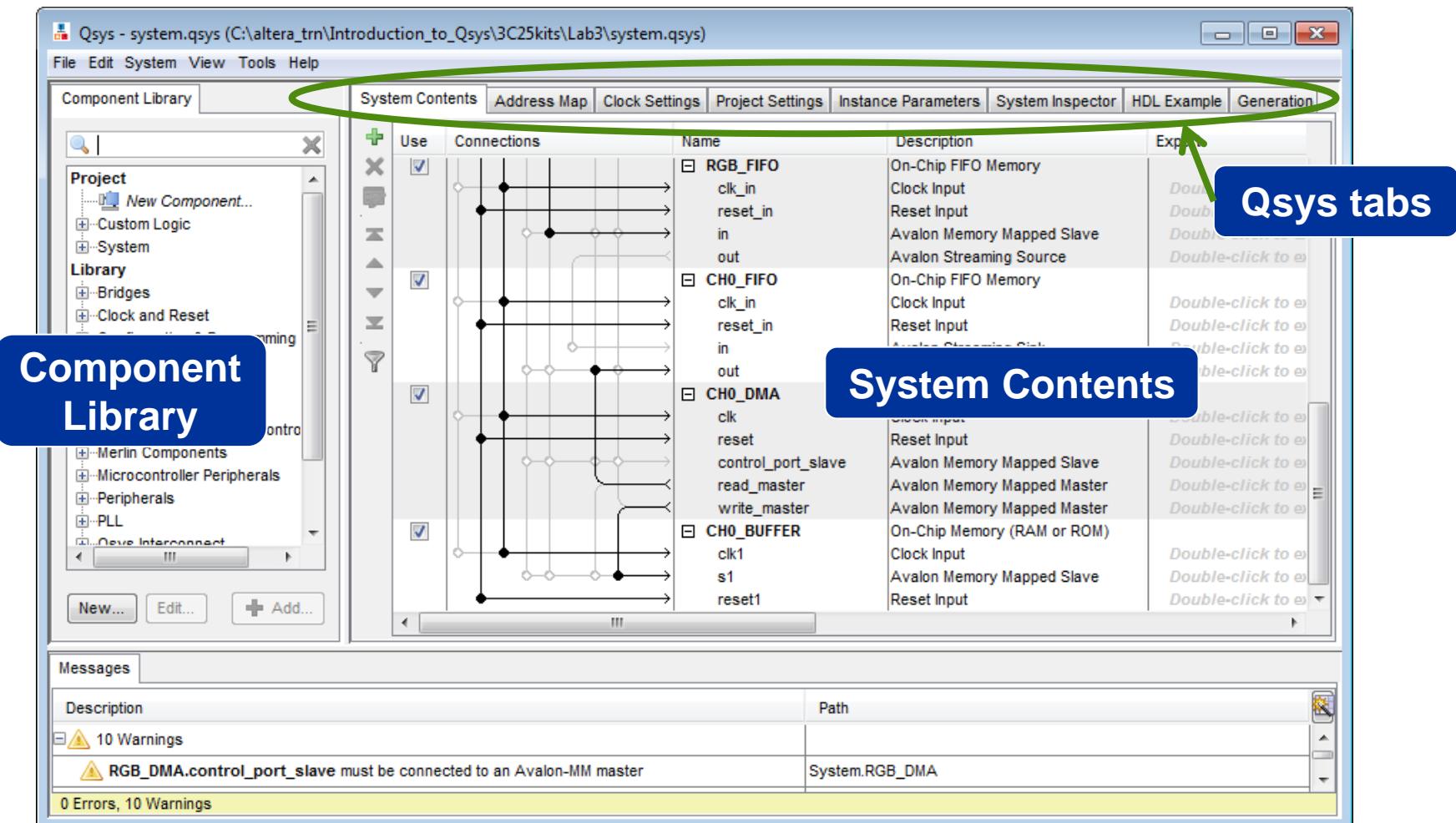
Automatic Interconnect Generation



- Avoids error-prone integration
- Saves development time with automatic logic & HDL generation
- Enables you to focus on value-add blocks

Qsys improves productivity by automatically generating the system interconnect logic

Qsys System-Integration Tool



[Introduction to the Qsys System Integration Tool](#) and [Advanced Qsys System Integration Tool Methodologies](#)
instructor led trainings available

■ Pre-made design blocks

■ Benefits

- Configurable, parameterized settings add flexibility & portability
- “Drop-in” support to accelerate design entry
- Pre-optimized for Altera architecture

■ Two versions

- Quartus II megafunctions
- Intellectual Property (IP) megafunctions

■ Free & installed with Quartus II software

- Non-encrypted functions written in AHDL (Altera HDL)
- HDL simulation models installed in Quartus II libraries

■ Two types

- Altera-specific megafunctions (begin with “ALT”)
- Library of parameterized modules (LPMs)

■ Examples

- Arithmetic
- On-chip RAM/ROM
- PLLs
- DDR/QDR/RLDRAM memory controllers

- **Must purchase license (except IP base suite)**
 - Logic for IP function is encrypted
- **Two types**
 - MegaCore IP – Developed by Altera
 - Altera Megafunctions Partner Program (AMPP) IP
- **All MegaCore functions & some AMPP functions support OpenCore® Plus feature**
 - Develop design using free version of core
 - HDL simulation models provided with IP
 - Generate time-limited configuration/programming files
 - See [AN320: OpenCore Plus Evaluation of Megafunctions](#)

■ Included in IP base suite

- FIR Compiler
- Fast Fourier Transform Compiler
- DDR3 SDRAM Controller with UniPHY

■ License required

- Triple-Speed Ethernet MAC
- 10Gb Ethernet MAC
- CRC Compiler
- IP Compiler for PCI Express® soft core
 - Note: PCI Express hard IP does not required a license
- Video and Image Processing Suite

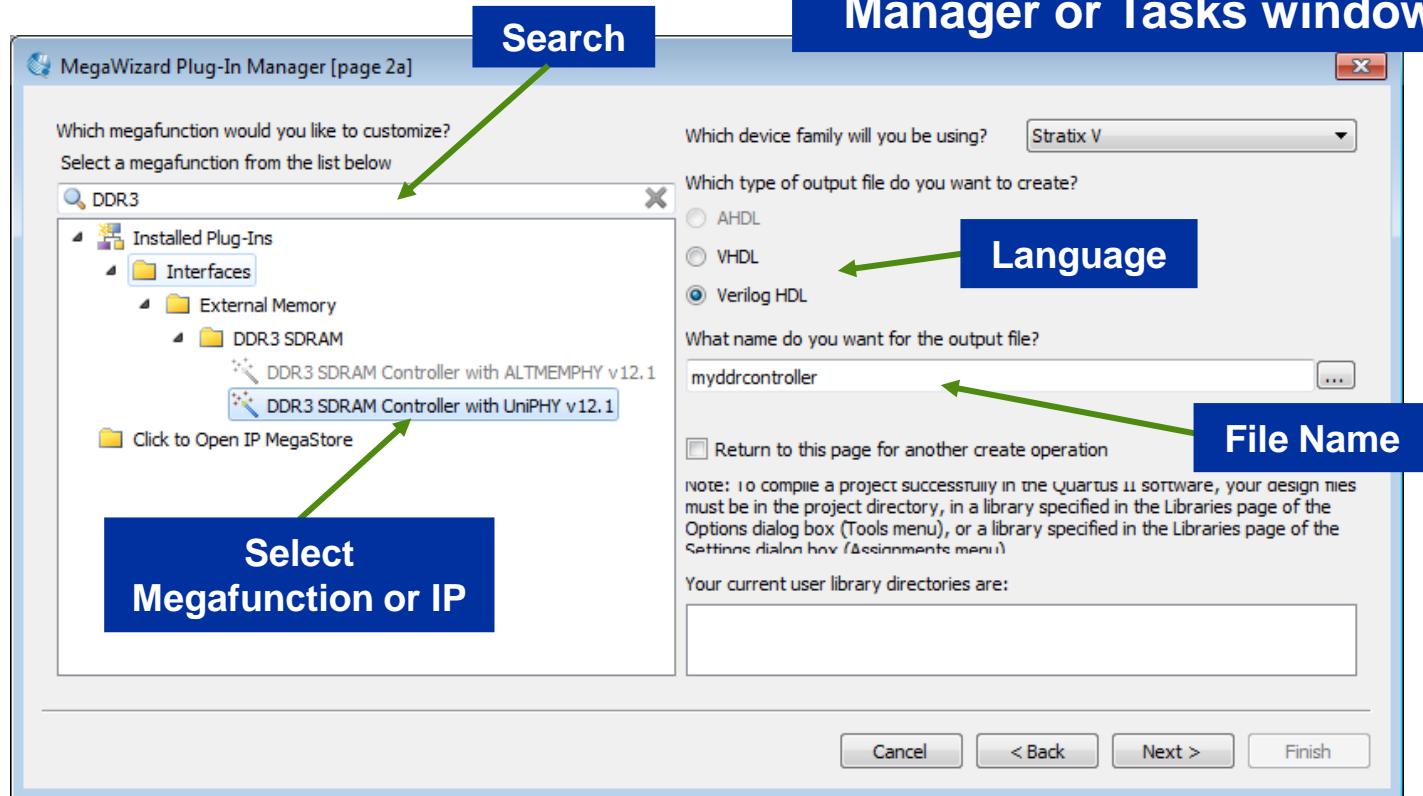
See <http://www.altera.com/products/ip/ipm-index.html> for a complete list of Altera IP solutions

MegaWizard Plug-in Manager



- Eases implementation and configuration of megafunctions & IP
- GUI, command line, or both

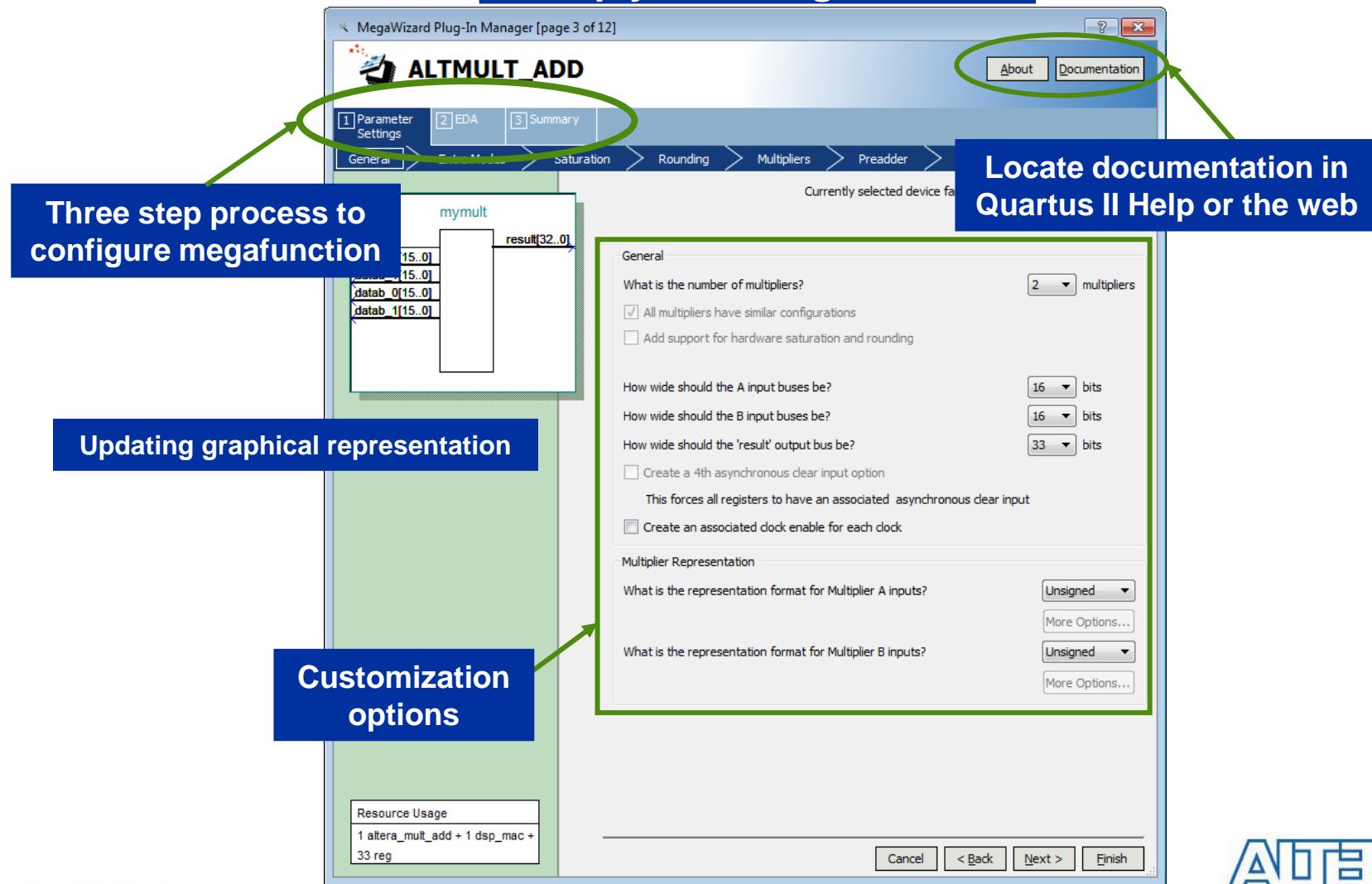
Tools → MegaWizard Plug-In Manager or Tasks window



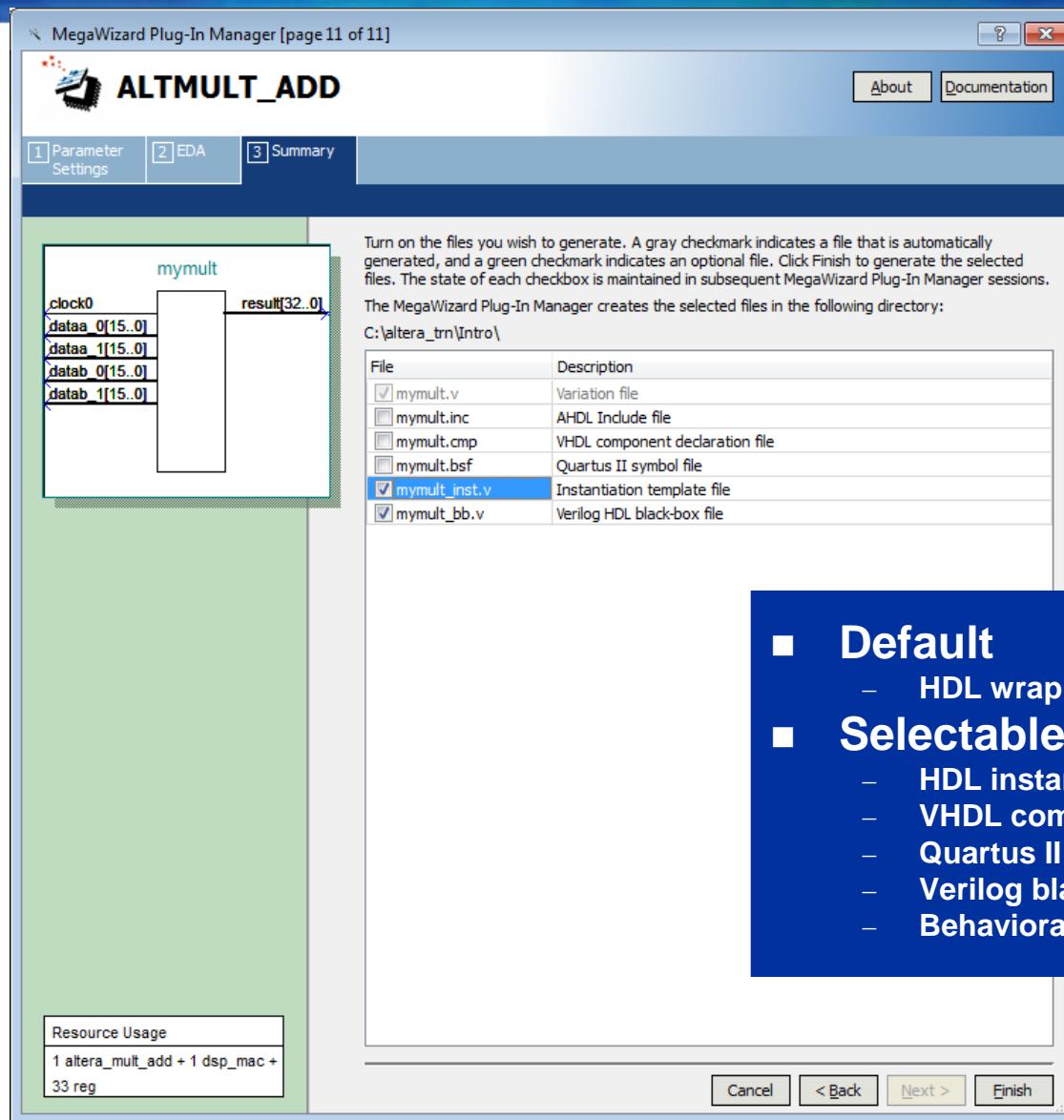
Command line: `qmegawiz <-silent> <module / wizard>=<mf_name> <ports & parameters options> file_name`

MegaWizard Example

Multiply-Add megafunction



MegaWizard Output File Selection



State Machine Editor

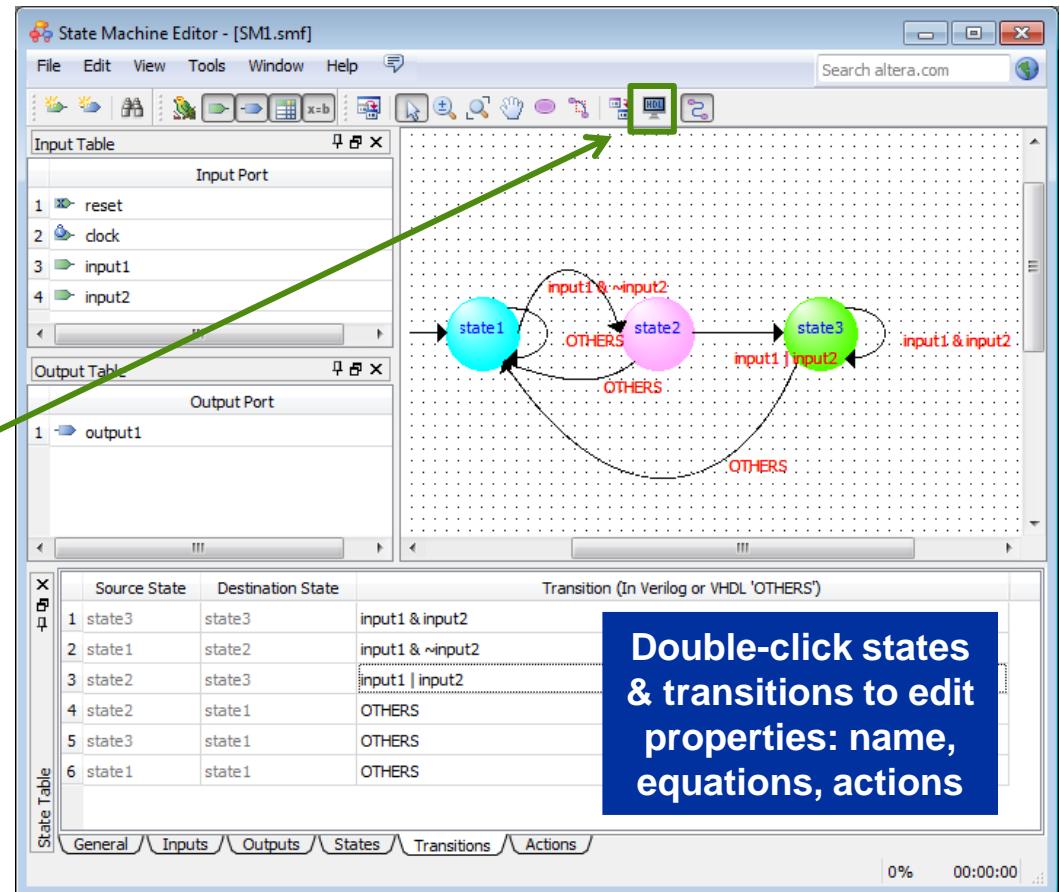
■ Create state machines in GUI

- Manually by adding individual states, transitions, and output actions
- Automatically with State Machine Wizard (Tools menu)

■ Generate state machine HDL code (required)

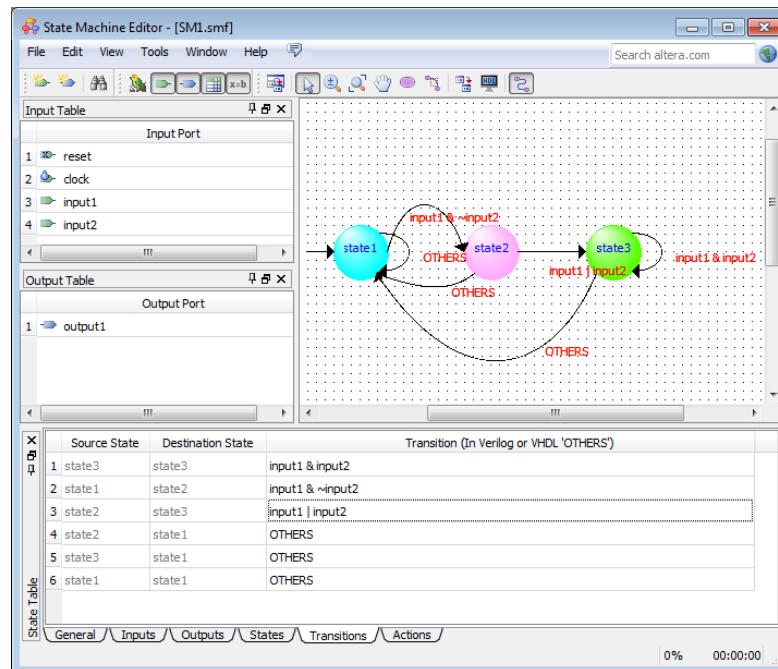
- VHDL
- Verilog
- SystemVerilog

File ⇒ New or Tasks window
Select *State Machine File (.SMF)*



From .SMF to HDL

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



```
abc Text Editor - [SM1.sv]
File Edit View Project Processing Tools Window Help > Search altera.com
File Edit View Project Processing Tools Window Help > Search altera.com
22 module SM1 (
23     input reset, input clock, input input1, input input2,
24     output output1);
25
26     reg reg_output1;
27     enum int unsigned { state1=0, state2=1, state3=2 };
28
29     always_ff @(posedge clock)
30     begin
31         if (clock) begin
32             fstate <= reg_fstate;
33         end
34     end
35
36     always_comb begin
37         if (reset) begin
38             reg_fstate <= state1;
39             reg_output1 <= 1'b0;
40             output1 <= 1'b0;
41         end
42         else begin
43             reg_output1 <= 1'b0;
44             output1 <= 1'b0;
45             case (fstate)
46                 state1: begin
47                     if ((input1 & ~input2))
48                         reg_fstate <= state2;
49                     else
50                         reg_fstate <= state1;
51
52                     reg_output1 <= 1'b0;
53                 end
54                 state2: begin
55                     if ((input1 & input2))
56                         reg_fstate <= state3;
57                     else
58                         reg_fstate <= state1;
59
60                     reg_output1 <= 1'b0;
61                 end
62                 state3: begin
63                     if ((input1 & ~input2))
64                         reg_fstate <= state1;
65                     else
66                         reg_fstate <= state3;
67
68                     reg_output1 <= 1'b0;
69                 end
70             endcase
71         end
72     end
73 endmodule
```

■ Create or edit memory initialization files

- Intel HEX (.HEX)
 - Preferred and compatible with 3rd party tools
- Altera-specific (.MIF) format

■ Design entry

- Initialization file data sent to device during device programming to initialize memory blocks

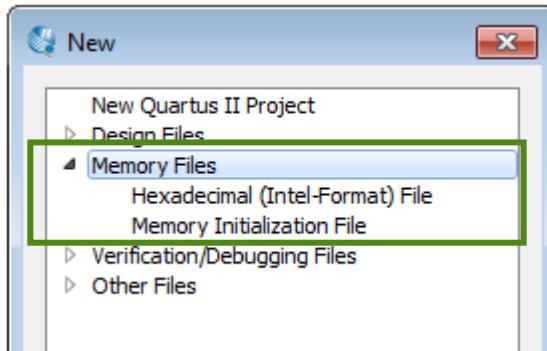
■ Simulation

- Use to initialize memory blocks before simulation or after breakpoints

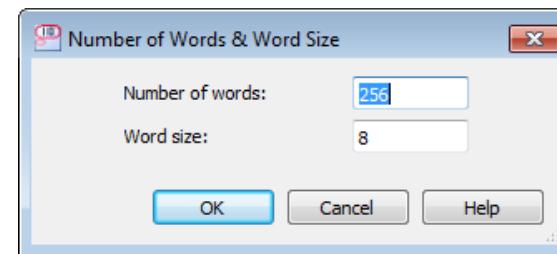
Create Memory Initialization File

File → New or Tasks window

1) HEX or MIF format



2) Select memory size



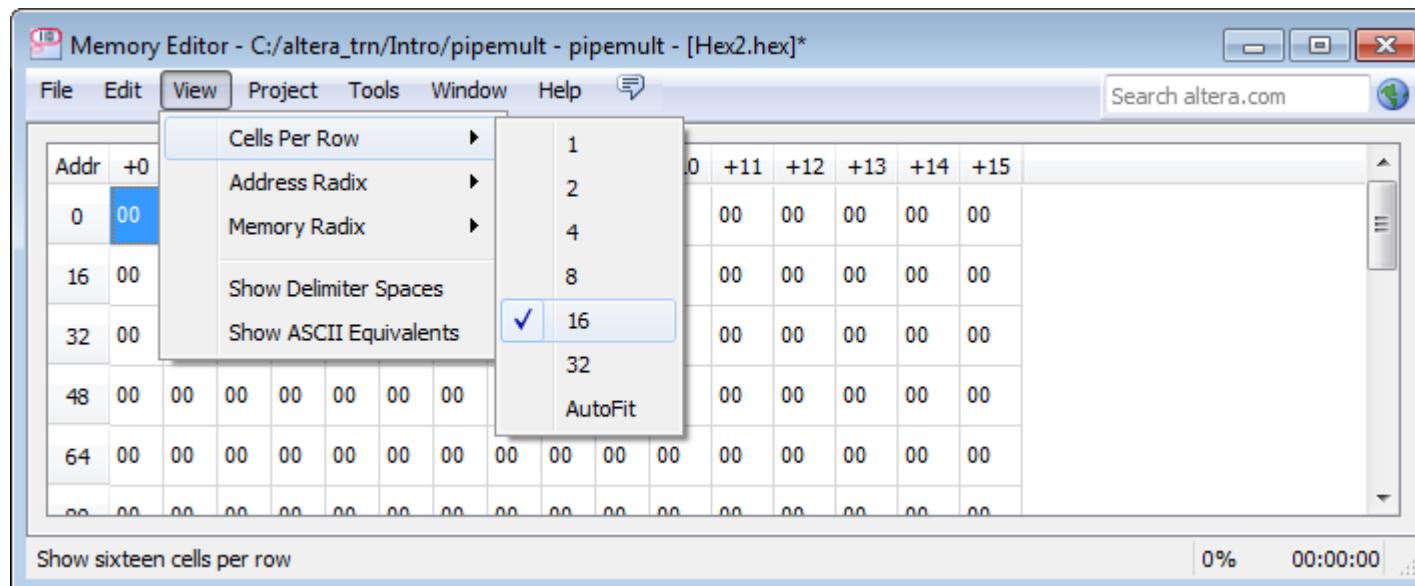
3) Memory space editor opens

Addr	+0	+1	+2	+3	+4	+5	+6	+7
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

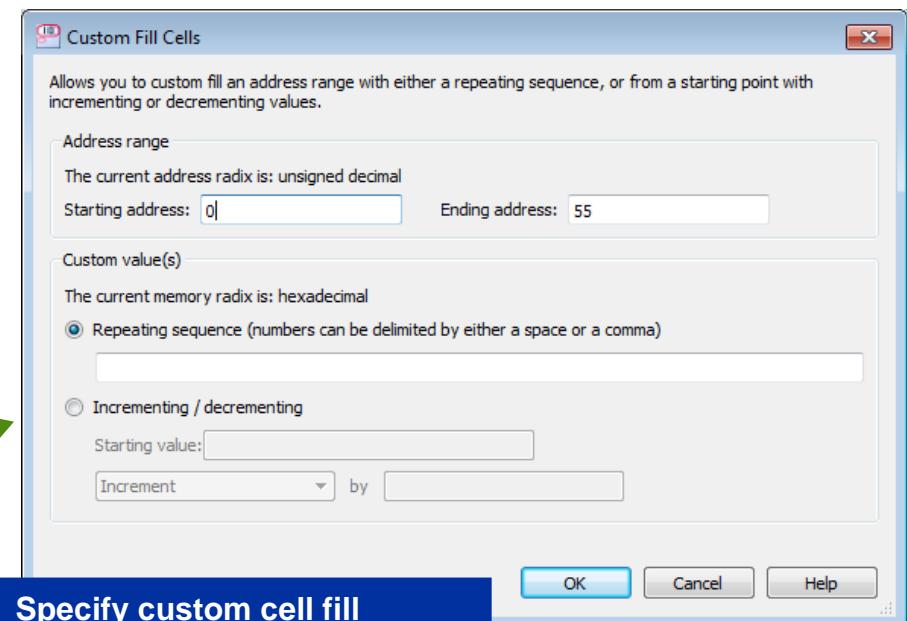
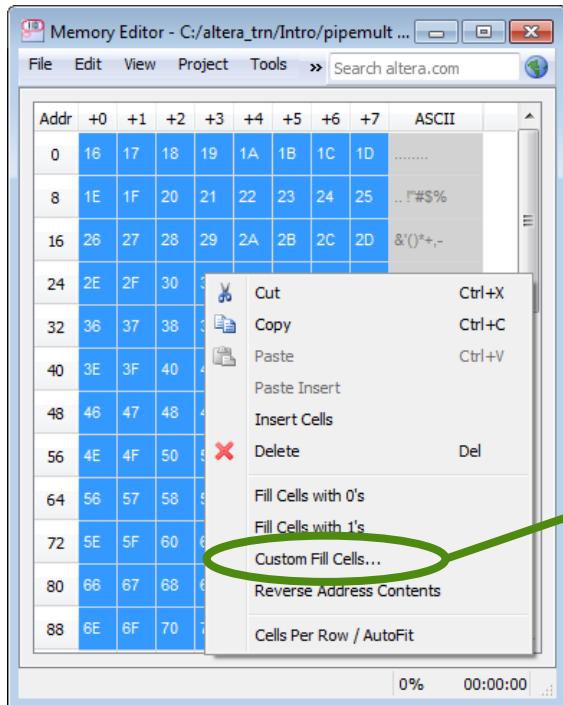
- View menu ⇒ select from available options



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

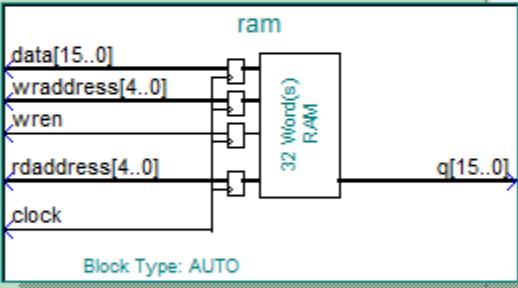
Using Memory File in Design

MegaWizard Plug-In Manager [page 8 of 10]

RAM: 2-PORT

1 Parameter Settings 2 EDA 3 Summary

General > Widths/Blk Type > Clks/Rd, Byte En > Regs/Ckens/Adrs > Output1 > Mem Init >



Do you want to specify the initial content of the memory?

No, leave it blank
 Initialize memory content data to XX..X on power-up in simulation

Yes, use this file for the memory content data
(You can use a Hexadecimal (Intel-format) File [.hex] or a Memory Initialization File [.mif])

Specify MIF or HEX file in MegaWizard Plug-In

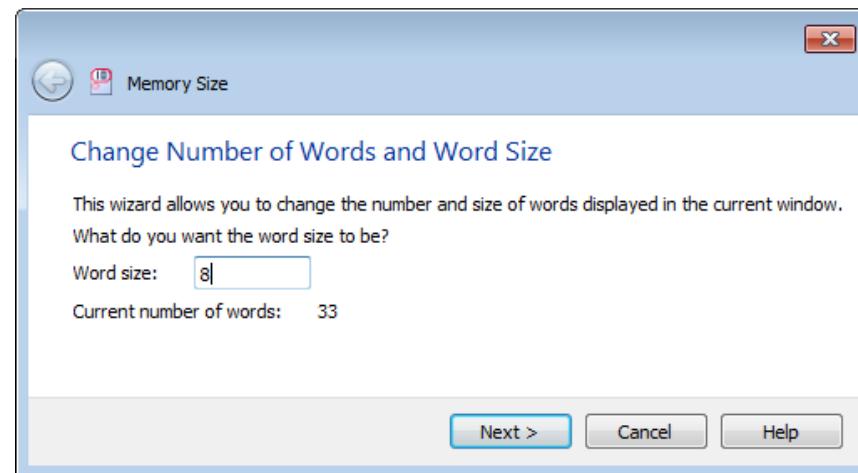
File name:

The initial content file should conform to which port's dimensions?

May also specify MIF file in HDL using the `ram_init_file` synthesis attribute

Memory Size Wizard

- Allows editing the size of 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 Quartus Mapping (.VQM)
- **To import and use netlist files**
 - Specify EDA tool in the Quartus II software settings
 - Instantiate block(s) in design
 - Add .EDF/.VQM file(s) to Quartus II project

3rd-Party Synthesis Tool Support



- LeonardoSpectrum
- Precision RTL



- Design Compiler
- Synplify

Test Your Knowledge: Design Entry

- 1. Name some of the design entry methods supported in the Quartus II software?**

A.

Exercise 2 Demonstration

*Demo should open automatically click
the link above if it doesn't*

Design Entry Summary

- **Multiple design entry methods supported**
 - Text (Verilog, VHDL, State Machine Editor output)
 - 3rd-party netlist (VQM, EDIF)
 - Schematic
 - System (Qsys)
- **MegaWizard Plug-In Manager parameterizes megafunctions & IP**
- **Memory Editor allows generation of memory initialization files**
- **3rd-party EDA tools supported for design entry & synthesis**

Design Entry Support Resources

■ Quartus II Handbook chapters

- [Recommended Design Practices](#) (Volume 1)
- [Recommended HDL Coding Styles](#) (Volume 1)
- 3rd-Party EDA tool chapters (Volume 1, Section 3)

■ 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 Quartus II Software](#)

Quartus II Software Design Series: Foundation

Quartus II Compilation



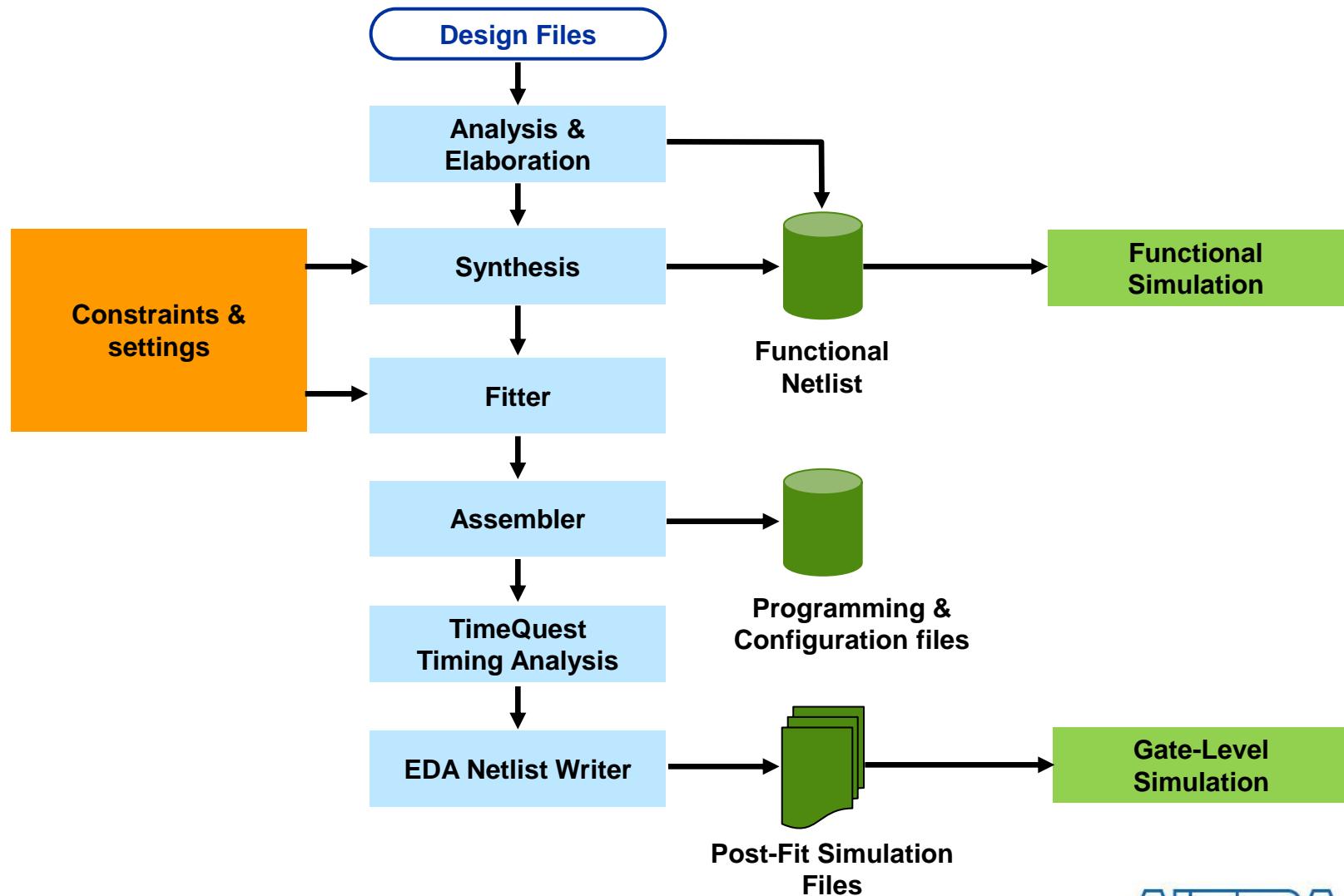
© 2013 Altera Corporation—Public



Compilation Section Objectives

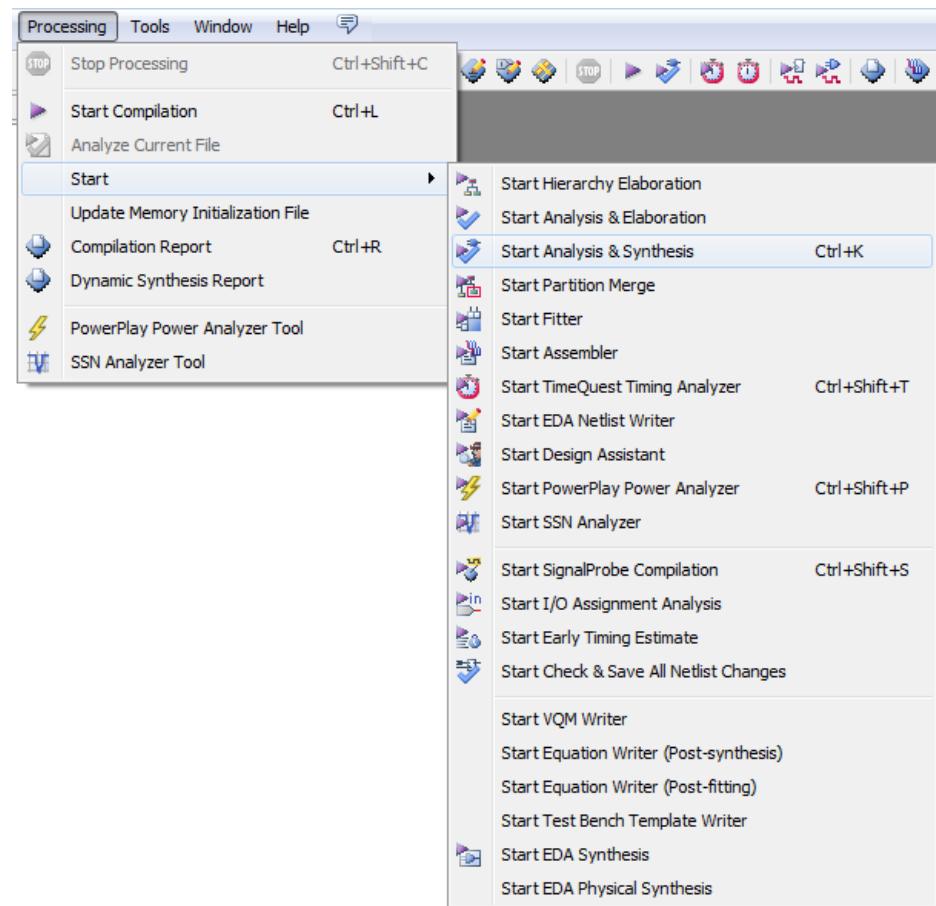
- **Describe the steps of the Quartus II compilation flow**
- **Utilize various Quartus II software tools used to understand how the design was processed**

Quartus II Full Compilation Flow



Processing Options

- **Start Compilation (Full)**
- **Start Hierarchy Elaboration**
 - Checks syntax & builds hierarchy
- **Start Analysis & Elaboration**
 - Checks syntax & builds hierarchy
 - Performs initial synthesis
- **Start Analysis & Synthesis**
 - Synthesizes & optimizes code
- **Start Fitter**
 - Places & routes design
- **Start Assembler**
 - Generate programming files
- **Analyzers**
 - I/O Assignment
 - PowerPlay
 - SSN (Switching Noise)
 - Design Assistant



Compilation Design Flows

■ Default “flat” compilation flow

- Design compiled as a whole
- Global optimizations performed

■ Incremental flow (on by default for new projects)

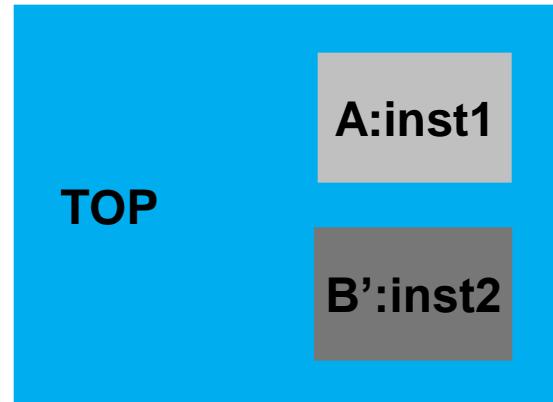
- User assigns design partitions
- Each partition processed separately; results merged
- Post-synthesis or post-fit netlists for partitions are reused
- Benefits
 - Decrease compilation time
 - Preserve compilation results and timing performance
 - Enable faster timing closure

Incremental Compilation Concept



$$\begin{matrix} + \\ \text{A} \\ + \\ \text{B}' \end{matrix}$$

=



Choose to reuse post-synthesis or post-fit netlist for TOP and A

Only specified portions of logic that have changed are re-synthesized or re-fitted

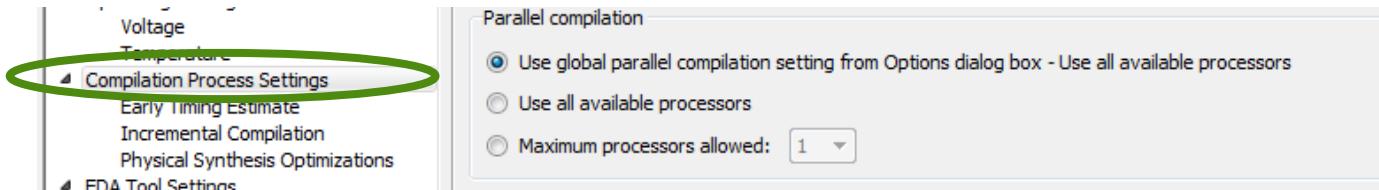
Note: For more details on using incremental compilation, please attend the course [Design Optimization Using Quartus II Incremental Compilation](#), watch the online training [Introduction to Incremental Compilation](#) or view the built-in interactive tutorial (Module 7: Incremental Compilation)

Reducing Compile Times

- **Incremental compilation**
- **Parallel compilation**

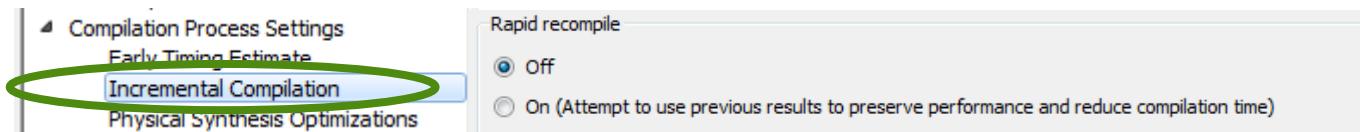
- Take advantage of multi-processor, multi-core machines

Assignments menu → Settings
(discussed in more detail later)

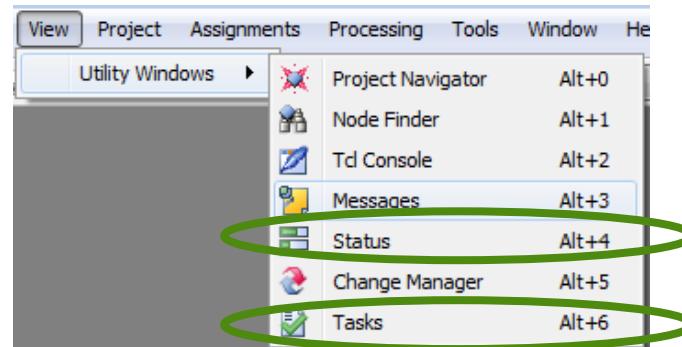


- **Rapid Recompile**

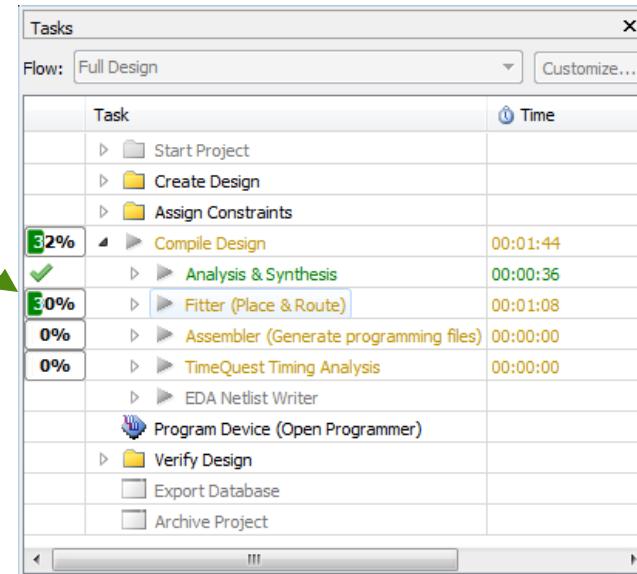
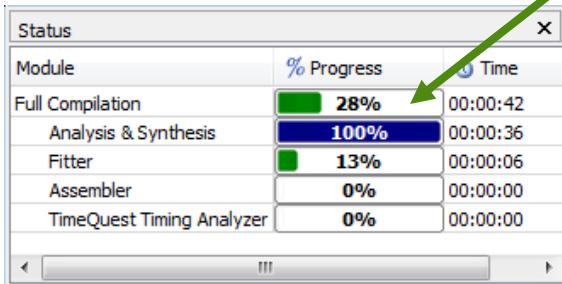
- Reuse compilation results to update design for small changes
- Works without incremental compilation



Status and Tasks Windows

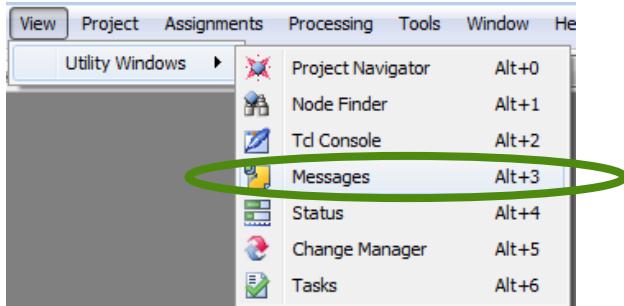


Status bars indicate compilation progress



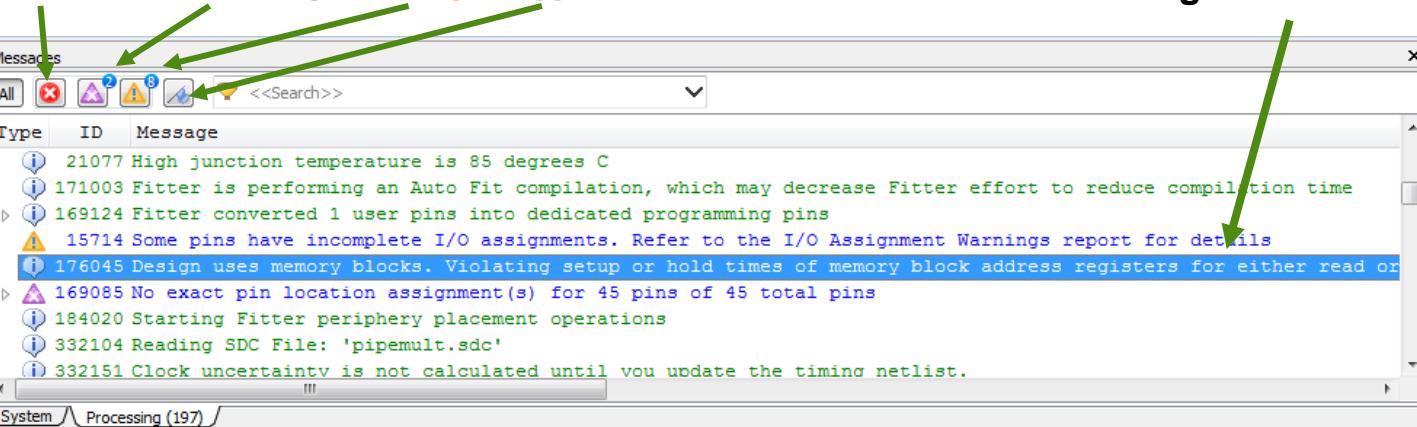
Message Windows

- Message window displays informational, warning, and error messages



Filter Messages by Type

All **Error** Critical Warning **Warning** Flagged

A screenshot of the 'Messages' window. At the top, there are filtering buttons for 'All', 'Error', 'Critical', 'Warning', and 'Flagged'. Below the buttons is a search bar with the placeholder '<<Search>>'. The main area is a table with columns 'Type', 'ID', and 'Message'. The table contains several messages, including:

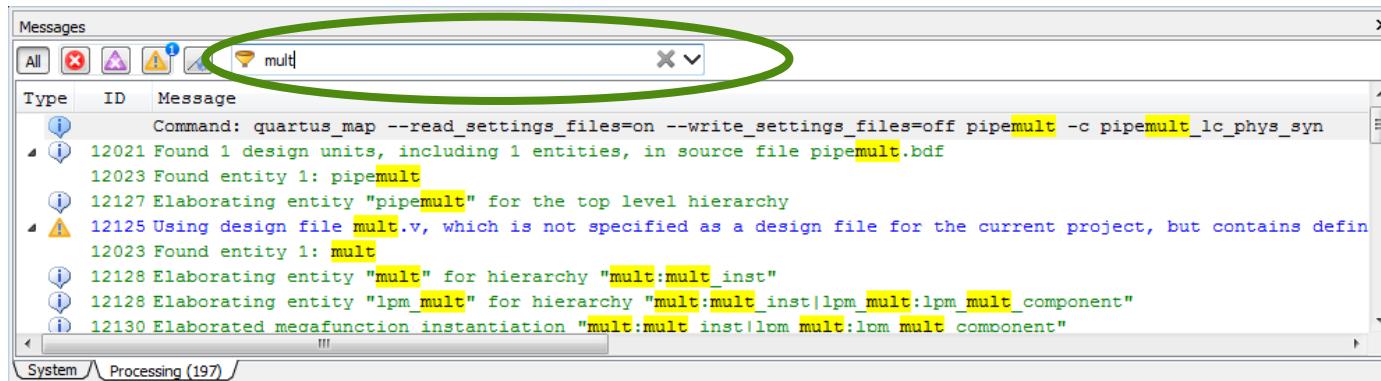
- 21077 High junction temperature is 85 degrees C
- 171003 Fitter is performing an Auto Fit compilation, which may decrease Fitter effort to reduce compilation time
- 169124 Fitter converted 1 user pins into dedicated programming pins
- 15714 Some pins have incomplete I/O assignments. Refer to the I/O Assignment Warnings report for details
- 176045 Design uses memory blocks. Violating setup or hold times of memory block address registers for either read or write can cause functional errors
- 169085 No exact pin location assignment(s) for 45 pins of 45 total pins
- 184020 Starting Fitter periphery placement operations
- 332104 Reading SDC File: 'pipemult.sdc'
- 332151 Clock uncertainty is not calculated until you update the timing netlist.

A specific message, '15714 Some pins have incomplete I/O assignments...', is highlighted with a blue selection bar. A green arrow points from the 'Flagged' filtering button to this highlighted message.

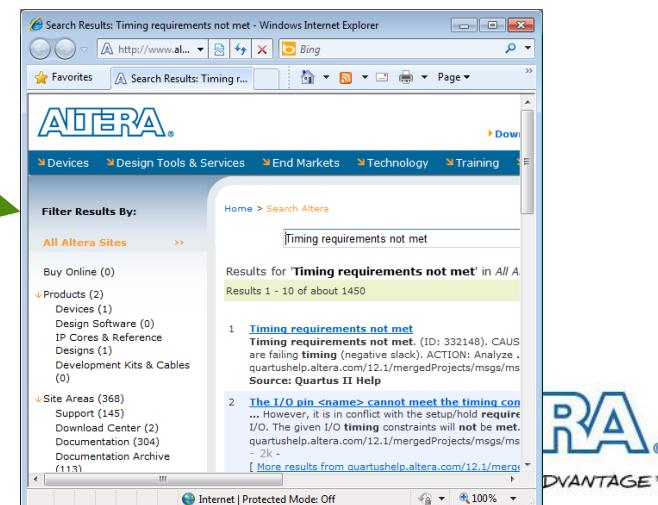
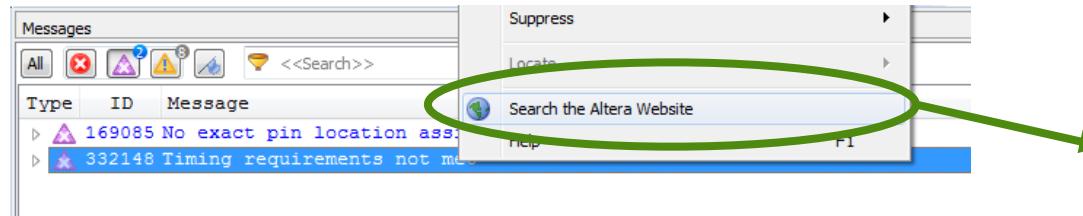
Right click to flag selected messages for later review

Message Searching

■ Search for keywords in the Messages window



■ Right click to search message on altera.com



Message IDs

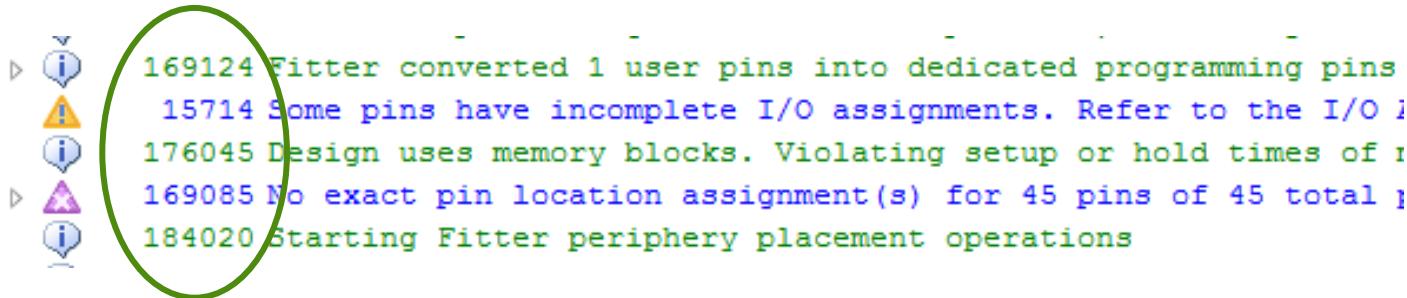
■ Most Quartus II messages have IDs

■ Benefits

- More easily reference messages in service requests
- Search for collateral documentation (Self-Help)
- Disable specific info or warning messages globally (MESSAGE_DISABLE)

■ Benefits for IP developers

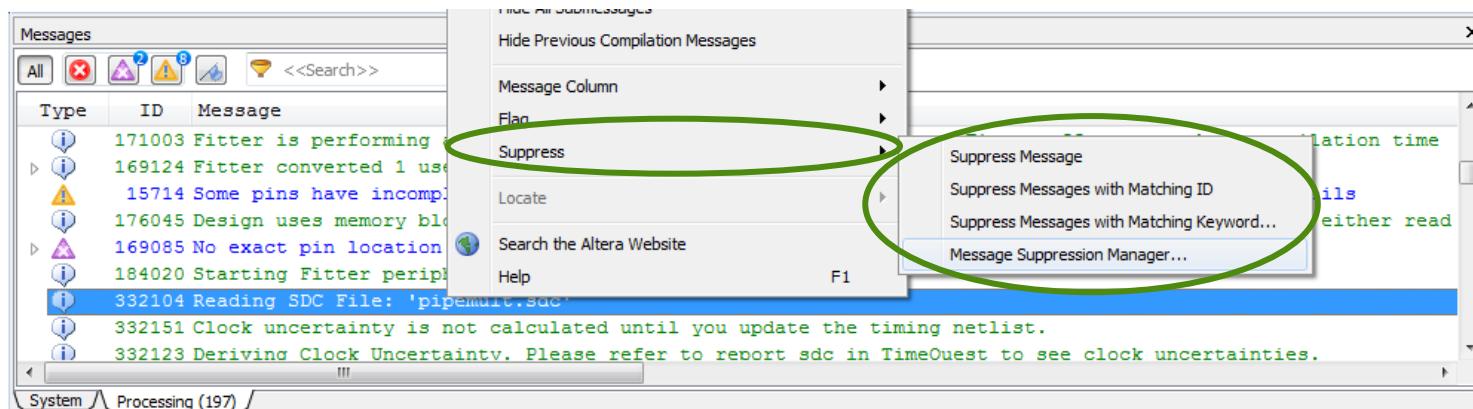
- Disable info or warning messages specifically for an IP core
- Ship “warning-free” IP



Message Suppression

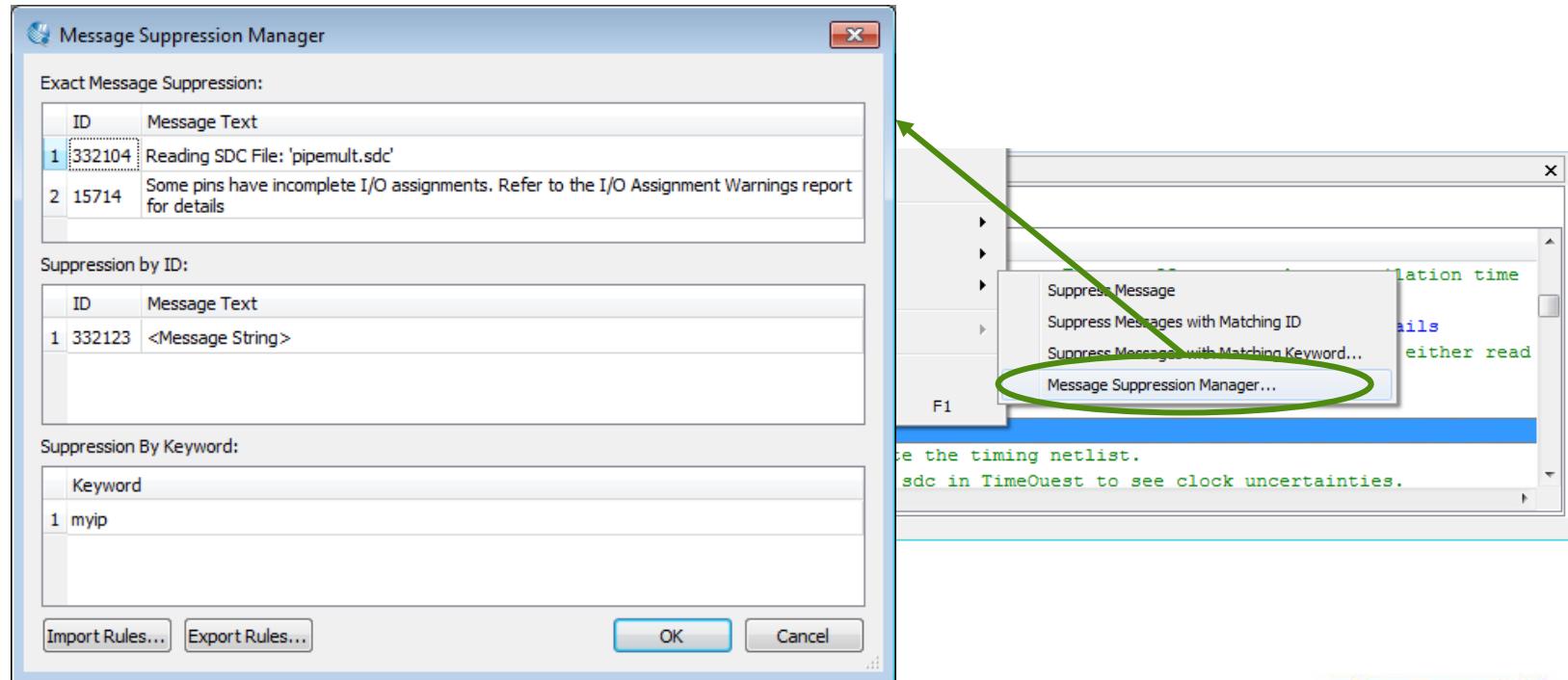
- **Hides messages from current and future compiles**
 - E.g. known synthesis warning message already investigated
- **Stores suppression rules in *<revision_name>.SRF* file**

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

■ Quartus II 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 project directory - *<revision_name>.fit.rpt*,
<revision_name>.map.rpt, etc

Compilation Report

■ Start from Processing menu or toolbar



Compilation Report - pipemult_lc_phys_syn

Table of Contents		Flow Summary																																																			
<ul style="list-style-type: none">Flow SummaryFlow SettingsFlow Non-Default Global SettingsFlow Elapsed TimeFlow OS SummaryFlow LogAnalysis & Synthesis<ul style="list-style-type: none">SummarySettingsParallel CompilationSource Files ReadResource Usage SummaryResource Utilization by EntityRAM SummaryDSP Block Usage SummaryIP Cores SummaryOptimization ResultsSource AssignmentsParameter Settings by Entity InstanceLPM Parameter SettingsElapsed Time Per PartitionMessagesFilterAssemblerTimeQuest Timing Analyzer	<table border="1"><thead><tr><th colspan="2">Flow Summary</th></tr></thead><tbody><tr><td>Flow Status</td><td>Successful - Thu Dec 20 18:48:58 2012</td></tr><tr><td>Quartus II 64-Bit Version</td><td>12.1 Build 177 11/07/2012 SJ Full Version</td></tr><tr><td>Revision Name</td><td>pipemult_lc_phys_syn</td></tr><tr><td>Top-level Entity Name</td><td>pipemult</td></tr><tr><td>Family</td><td>Stratix V</td></tr><tr><td>Device</td><td>5SEEBF45C4</td></tr><tr><td>Timing Models</td><td>Preliminary</td></tr><tr><td>Logic utilization (in ALMs)</td><td>14 / 359,200 (< 1 %)</td></tr><tr><td>Total registers</td><td>22</td></tr><tr><td>Total pins</td><td>45 / 840 (5 %)</td></tr><tr><td>Total virtual pins</td><td>0</td></tr><tr><td>Total block memory bits</td><td>0 / 54,067,200 (0 %)</td></tr><tr><td>Total DSP Blocks</td><td>1 / 352 (< 1 %)</td></tr><tr><td>Total HSSI STD RX PCSS</td><td>0</td></tr><tr><td>Total HSSI 10G RX PCSS</td><td>0</td></tr><tr><td>Total HSSI GEN3 RX PCSS</td><td>0</td></tr><tr><td>Total HSSI PMA RX Deserializers</td><td>0</td></tr><tr><td>Total HSSI STD TX PCSS</td><td>0</td></tr><tr><td>Total HSSI 10G TX PCSS</td><td>0</td></tr><tr><td>Total HSSI GEN3 TX PCSS</td><td>0</td></tr><tr><td>Total HSSI PMA TX Serializers</td><td>0</td></tr><tr><td>Total HSSI PIPE GEN1_2s</td><td>0</td></tr><tr><td>Total HSSI GEN3s</td><td>0</td></tr><tr><td>Total PLLs</td><td>0 / 50 (0 %)</td></tr><tr><td>Total DLLs</td><td>0 / 4 (0 %)</td></tr></tbody></table>	Flow Summary		Flow Status	Successful - Thu Dec 20 18:48:58 2012	Quartus II 64-Bit Version	12.1 Build 177 11/07/2012 SJ Full Version	Revision Name	pipemult_lc_phys_syn	Top-level Entity Name	pipemult	Family	Stratix V	Device	5SEEBF45C4	Timing Models	Preliminary	Logic utilization (in ALMs)	14 / 359,200 (< 1 %)	Total registers	22	Total pins	45 / 840 (5 %)	Total virtual pins	0	Total block memory bits	0 / 54,067,200 (0 %)	Total DSP Blocks	1 / 352 (< 1 %)	Total HSSI STD RX PCSS	0	Total HSSI 10G RX PCSS	0	Total HSSI GEN3 RX PCSS	0	Total HSSI PMA RX Deserializers	0	Total HSSI STD TX PCSS	0	Total HSSI 10G TX PCSS	0	Total HSSI GEN3 TX PCSS	0	Total HSSI PMA TX Serializers	0	Total HSSI PIPE GEN1_2s	0	Total HSSI GEN3s	0	Total PLLs	0 / 50 (0 %)	Total DLLs	0 / 4 (0 %)
Flow Summary																																																					
Flow Status	Successful - Thu Dec 20 18:48:58 2012																																																				
Quartus II 64-Bit Version	12.1 Build 177 11/07/2012 SJ Full Version																																																				
Revision Name	pipemult_lc_phys_syn																																																				
Top-level Entity Name	pipemult																																																				
Family	Stratix V																																																				
Device	5SEEBF45C4																																																				
Timing Models	Preliminary																																																				
Logic utilization (in ALMs)	14 / 359,200 (< 1 %)																																																				
Total registers	22																																																				
Total pins	45 / 840 (5 %)																																																				
Total virtual pins	0																																																				
Total block memory bits	0 / 54,067,200 (0 %)																																																				
Total DSP Blocks	1 / 352 (< 1 %)																																																				
Total HSSI STD RX PCSS	0																																																				
Total HSSI 10G RX PCSS	0																																																				
Total HSSI GEN3 RX PCSS	0																																																				
Total HSSI PMA RX Deserializers	0																																																				
Total HSSI STD TX PCSS	0																																																				
Total HSSI 10G TX PCSS	0																																																				
Total HSSI GEN3 TX PCSS	0																																																				
Total HSSI PMA TX Serializers	0																																																				
Total HSSI PIPE GEN1_2s	0																																																				
Total HSSI GEN3s	0																																																				
Total PLLs	0 / 50 (0 %)																																																				
Total DLLs	0 / 4 (0 %)																																																				

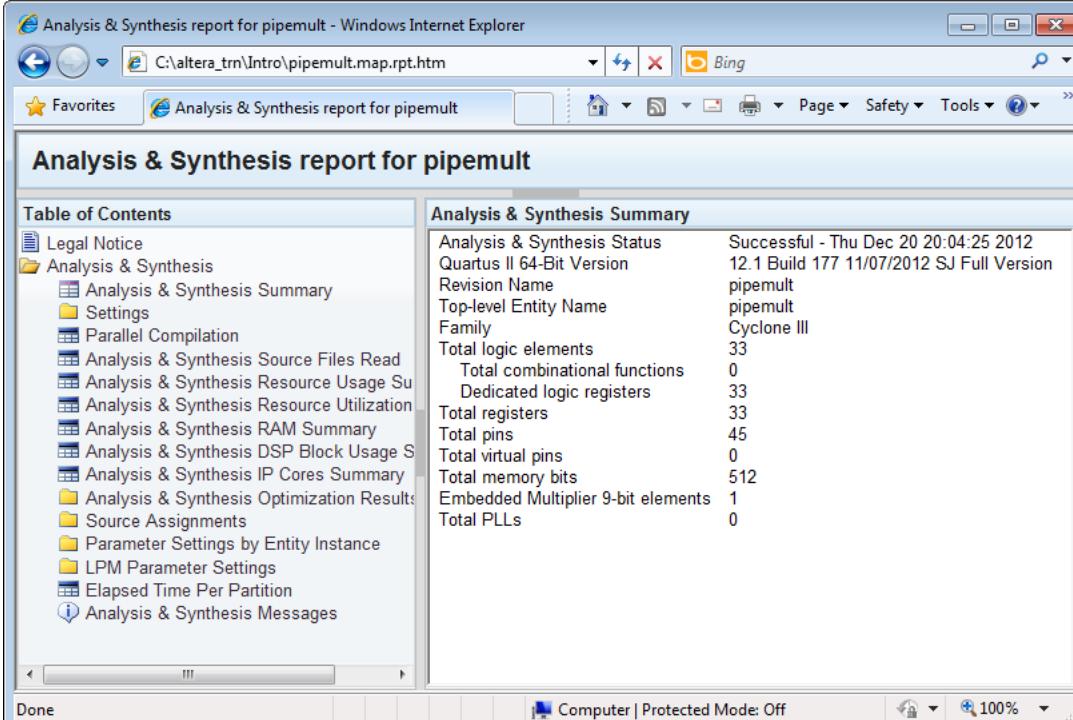
Each compiler executable generates separate folder

Compilation report - HTML version

■ Optional HTML Report

- Enable from Tools->Options->Processing->Automatically generate HTML-Format Report Files after design processing

■ View in any web browser

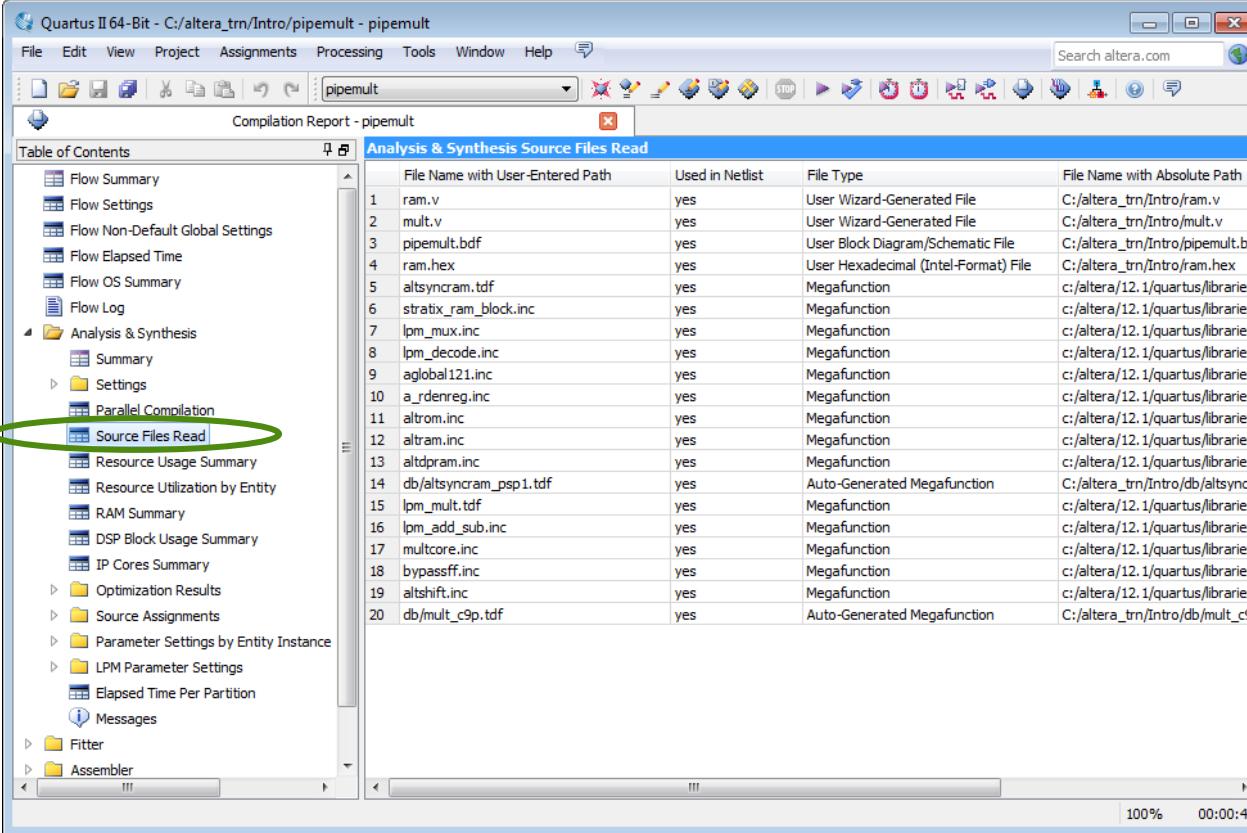


The screenshot shows a Windows Internet Explorer window with the title "Analysis & Synthesis report for pipemult - Windows Internet Explorer". The address bar shows the URL "C:\altera_trn\Intro\pipemult.map.rpt.htm". The page content is titled "Analysis & Synthesis report for pipemult". On the left, there is a "Table of Contents" sidebar with a hierarchical list of synthesis reports. On the right, there is a "Analysis & Synthesis Summary" table with the following data:

Analysis & Synthesis Summary	
Analysis & Synthesis Status	Successful - Thu Dec 20 20:04:25 2012
Quartus II 64-Bit Version	12.1 Build 177 11/07/2012 SJ Full Version
Revision Name	pipemult
Top-level Entity Name	pipemult
Family	Cyclone III
Total logic elements	33
Total combinational functions	0
Dedicated logic registers	33
Total registers	33
Total pins	45
Total virtual pins	0
Total memory bits	512
Embedded Multiplier 9-bit elements	1
Total PLLs	0

Example: Source Files Read

- **Source Files Read table lists all design files (user-coded & library) used during last compilation along with files' type and location**



The screenshot shows the Quartus II 64-Bit software interface with the project 'C:/altera_trn/Intro/pipemult' open. The 'Analysis & Synthesis Source Files Read' table is displayed, listing 20 files with their details. The 'Source Files Read' item in the Table of Contents is circled in green.

File Name with User-Entered Path	Used in Netlist	File Type	File Name with Absolute Path
1 ram.v	yes	User Wizard-Generated File	C:/altera_trn/Intro/ram.v
2 mult.v	yes	User Wizard-Generated File	C:/altera_trn/Intro/mult.v
3 pipemult.bdf	yes	User Block Diagram/Schematic File	C:/altera_trn/Intro/pipemult.bdf
4 ram.hex	yes	User Hexadecimal (Intel-Format) File	C:/altera_trn/Intro/ram.hex
5 altsyncram.tdf	yes	Megafunction	c:/altera/12.1/quartus/libraries
6 stratix_ram_block.inc	yes	Megafunction	c:/altera/12.1/quartus/libraries
7 lpm_mux.inc	yes	Megafunction	c:/altera/12.1/quartus/libraries
8 lpm_decode.inc	yes	Megafunction	c:/altera/12.1/quartus/libraries
9 aglobal121.inc	yes	Megafunction	c:/altera/12.1/quartus/libraries
10 a_rdreneg.inc	yes	Megafunction	c:/altera/12.1/quartus/libraries
11 altrom.inc	yes	Megafunction	c:/altera/12.1/quartus/libraries
12 altram.inc	yes	Megafunction	c:/altera/12.1/quartus/libraries
13 altdpram.inc	yes	Megafunction	c:/altera/12.1/quartus/libraries
14 db/altsyncram_psp1.tdf	yes	Auto-Generated Megafunction	C:/altera_trn/Intro/db/altsync
15 lpm_mult.tdf	yes	Megafunction	c:/altera/12.1/quartus/libraries
16 lpm_add_sub.inc	yes	Megafunction	c:/altera/12.1/quartus/libraries
17 multcore.inc	yes	Megafunction	c:/altera/12.1/quartus/libraries
18 bypassff.inc	yes	Megafunction	c:/altera/12.1/quartus/libraries
19 altshift.inc	yes	Megafunction	c:/altera/12.1/quartus/libraries
20 db/mult_c9p.tdf	yes	Auto-Generated Megafunction	C:/altera_trn/Intro/db/mult_c9

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

The screenshot shows the Quartus II Compilation Report interface. The left pane displays the 'Table of Contents' for the 'Compilation Report - pipemult'. The 'Analysis & Synthesis Resource Usage' section is expanded, showing various resource categories and their details. The 'Fitter Resource Usage Summary' section is also visible on the right, showing detailed usage statistics for logic elements, registers, and other resources.

Analysis & Synthesis Resource Usage

- Resource
- 1 Estimated Total logic elements
- 2
- 3 Total combinational functions
- 4 Logic element usage by number of inputs
 - 1 -- 4 input functions
 - 2 -- 3 input functions
 - 3 -- <=2 input functions
- 5
- 6 Logic elements by mode
 - 1 -- normal mode
 - 2 -- arithmetic mode
- 7
- 8 Total registers
 - 1 -- Dedicated logic registers
 - 2 -- I/O registers
- 9
- 10 I/O pins
- 11 Total memory bits
- 12 Embedded Multiplier 9-bit elements
- 13 Maximum fan-out
- 14 Total fan-out
- 15 Average fan-out

Fitter Resource Usage Summary

Resource	Usage
1 Total logic elements	1 / 5,136 (< 1 %)
1 -- Combinational with no register	0
2 -- Register only	1
3 -- Combinational with a register	0
2	
3 Logic element usage by number of LUT inputs	
1 -- 4 input functions	0
2 -- 3 input functions	0
3 -- <=2 input functions	0
4 -- Register only	1
4	
5 Logic elements by mode	
1 -- normal mode	0
2 -- arithmetic mode	0
6	
7 Total registers*	17 / 6,000 (< 1 %)
1 -- Dedicated logic registers	1 / 5,136 (< 1 %)
2 -- I/O registers	16 / 864 (2 %)
8	
9 Total LABs: partially or completely used	1 / 321 (< 1 %)
10 Virtual pins	0
11 I/O pins	45 / 183 (25 %)
1 -- Clock pins	2 / 4 (50 %)
2 -- Dedicated input pins	0 / 9 (0 %)
12	
13 Global signals	2
14 M9Ks	1 / 46 (2 %)
15 Total block memory bits	512 / 423,936 (< 1 %)
16 Total block memory implementation bits	9,216 / 423,936 (2 %)
17 Embedded Multiplier 9-bit elements	1 / 46 (2 %)
18 PLLs	0 / 2 (0 %)
19 Global clocks	2 / 10 (20 %)

* Register count does not include registers inside RAM blocks or DSP blocks.

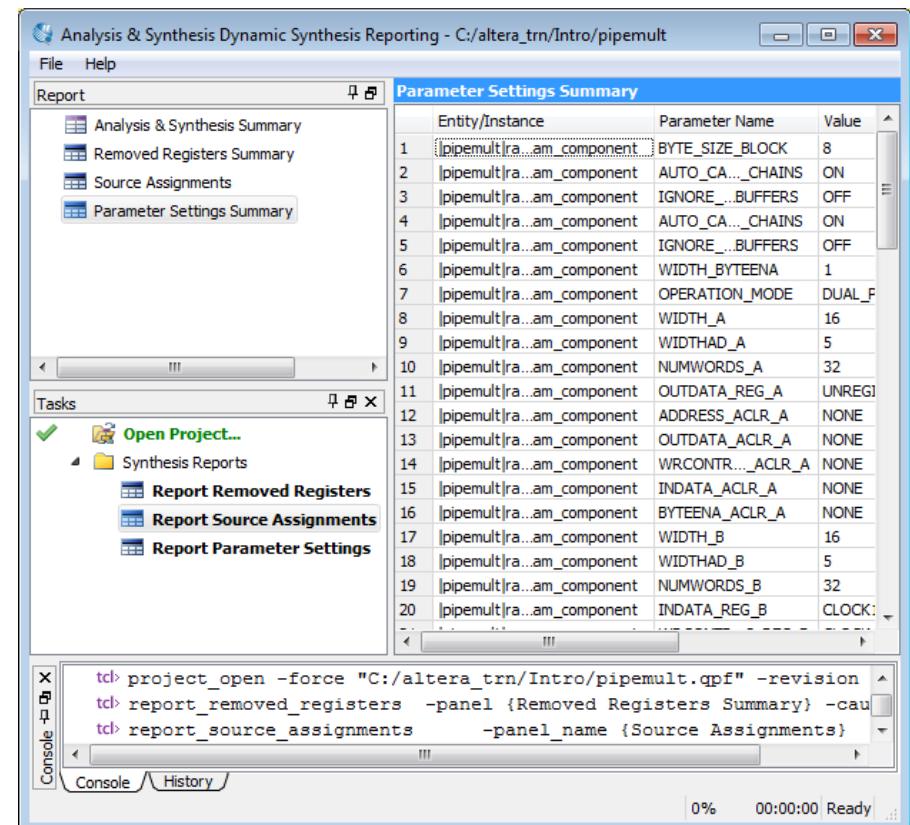
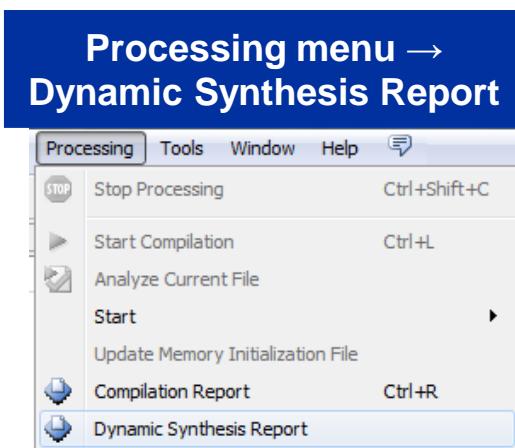
Dynamic Synthesis Report

■ Task/Report Model

- Customized entity/node filtered reports

■ Available Tasks

- Report Removed Registers
- Report Source Assignments
- Report Parameter Settings



■ RTL Viewer

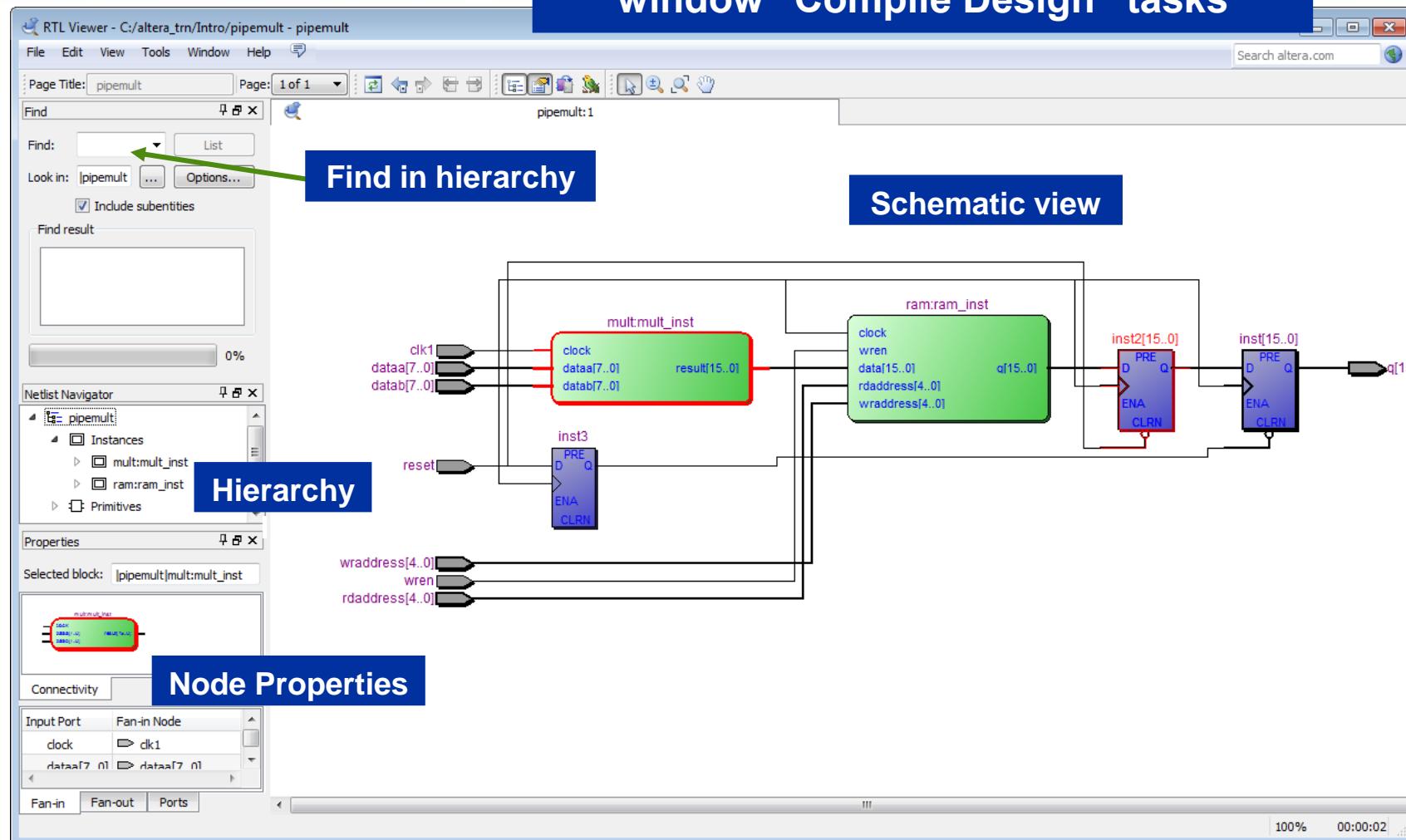
- Schematic of design after Analysis and Elaboration
- Visually check initial HDL before synthesis optimizations
- Locate synthesized nodes for assigning constraints
- Debug verification issues

■ 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)

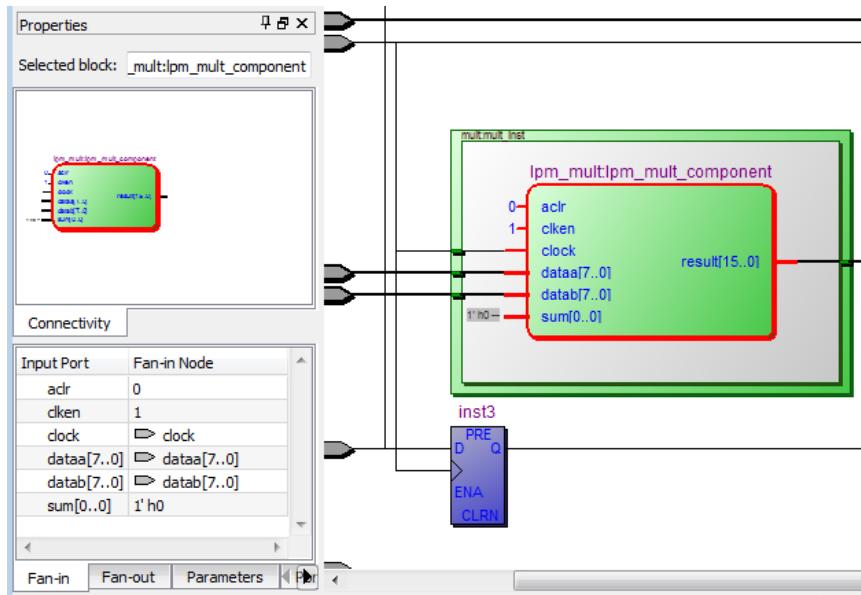
RTL Viewer

Tools menu → Netlist Viewers or Tasks
window “Compile Design” tasks



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

Schematic View (RTL Viewer)



Click on node to see details in the Properties window

■ Represents design using logic blocks & nets

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

Schematic Hierarchy Navigation

■ Descend hierarchy

- Double-click on instance
- Right-click & select **Hierarchy Down**



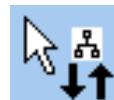
■ Ascend hierarchy

- Double-click in white space
- Right-click & select **Hierarchy Up**



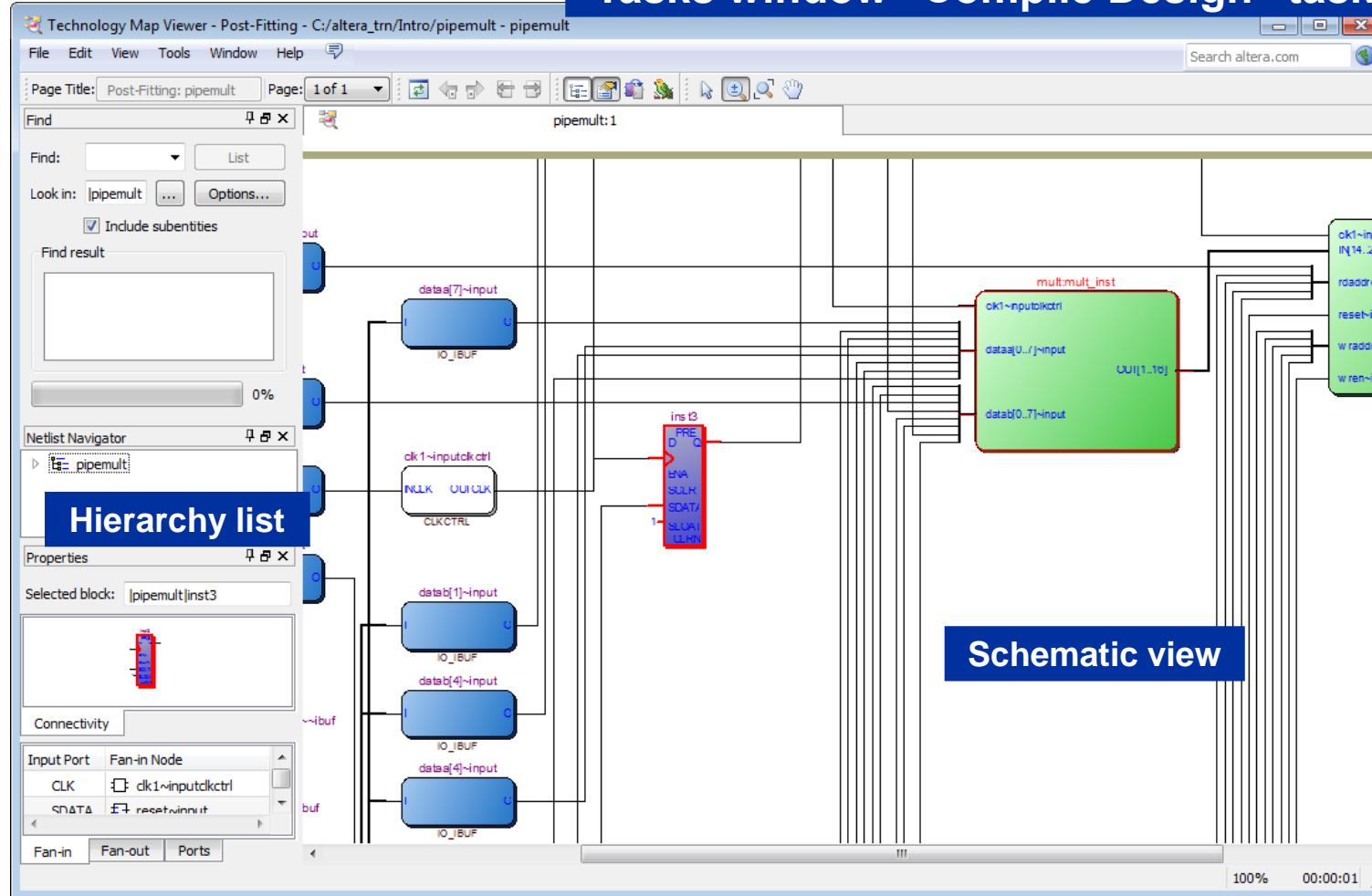
■ Middle hierarchy

- Double-click on instance descends
- Double-click in white space ascends



Technology Map Viewers

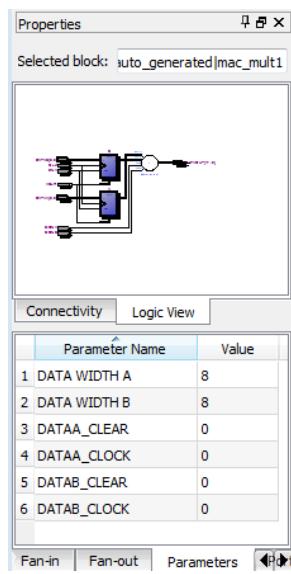
Tools Menu → Netlist Viewers or
Tasks window “Compile Design” tasks



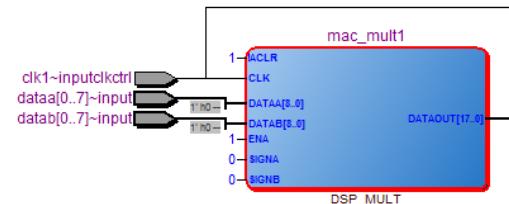
Note: Must run synthesis and/or fitting first

Schematic View (Technology Viewer)

- Represents design using atoms
 - I/O pins & cells
 - Lcells
 - Memory blocks
 - MAC (DSP blocks)

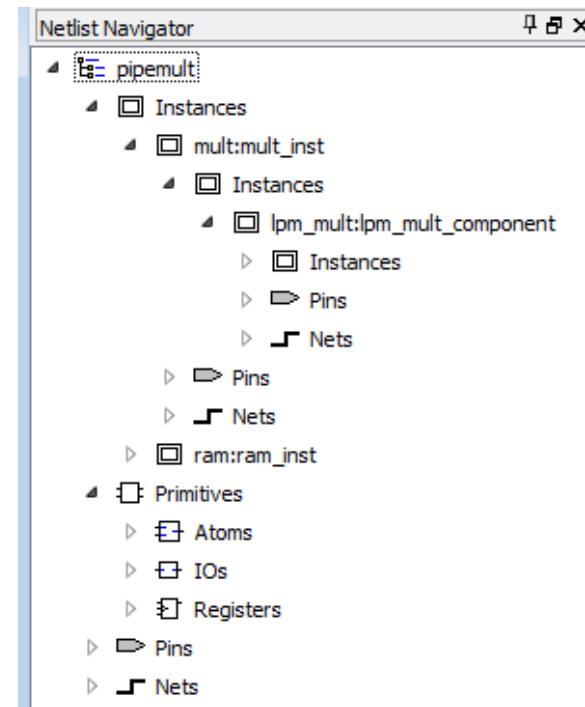


Click on node to see details in the properties window



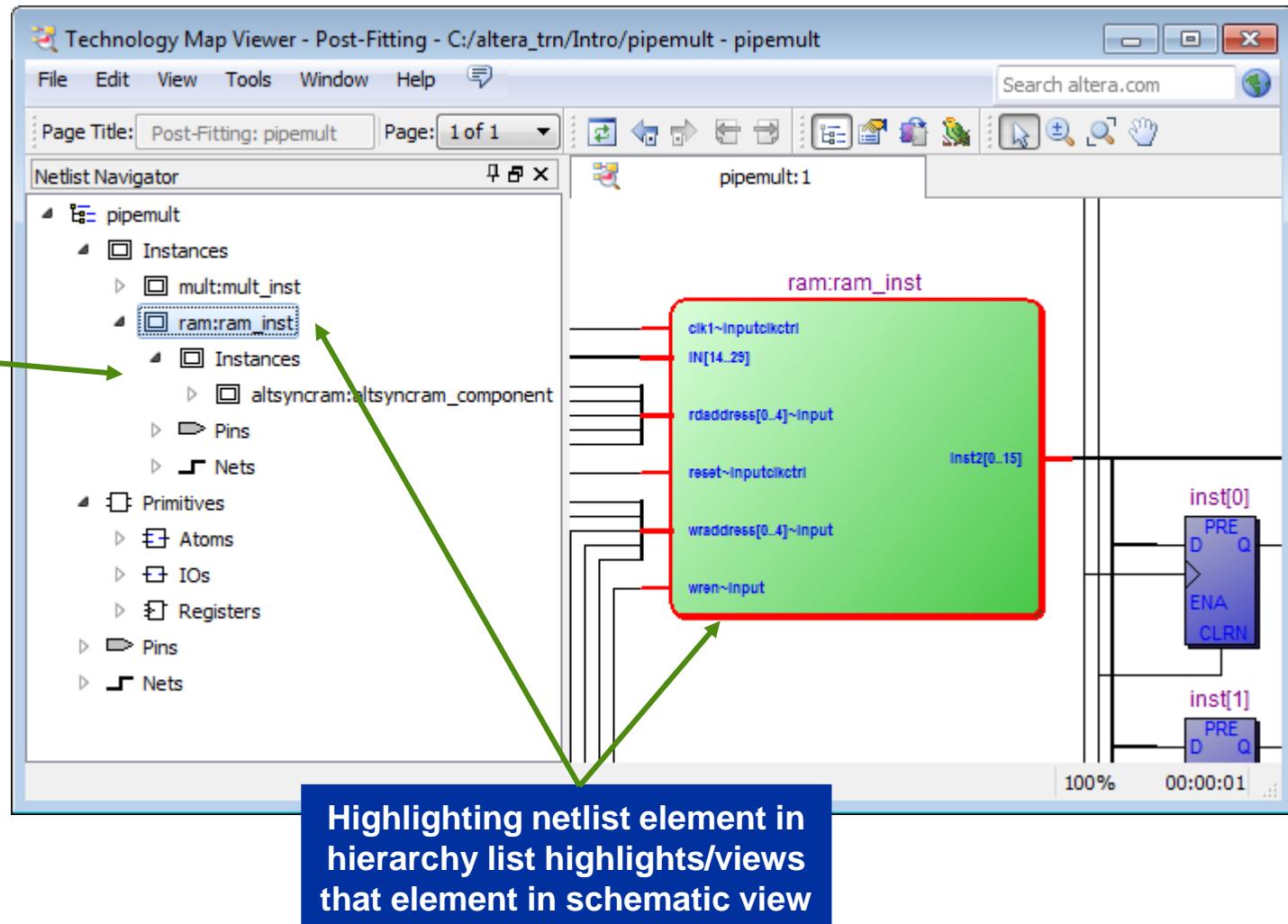
Hierarchy List

- 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
 - Pins
 - Nets
 - State machines
 - Logic clouds (if enabled)



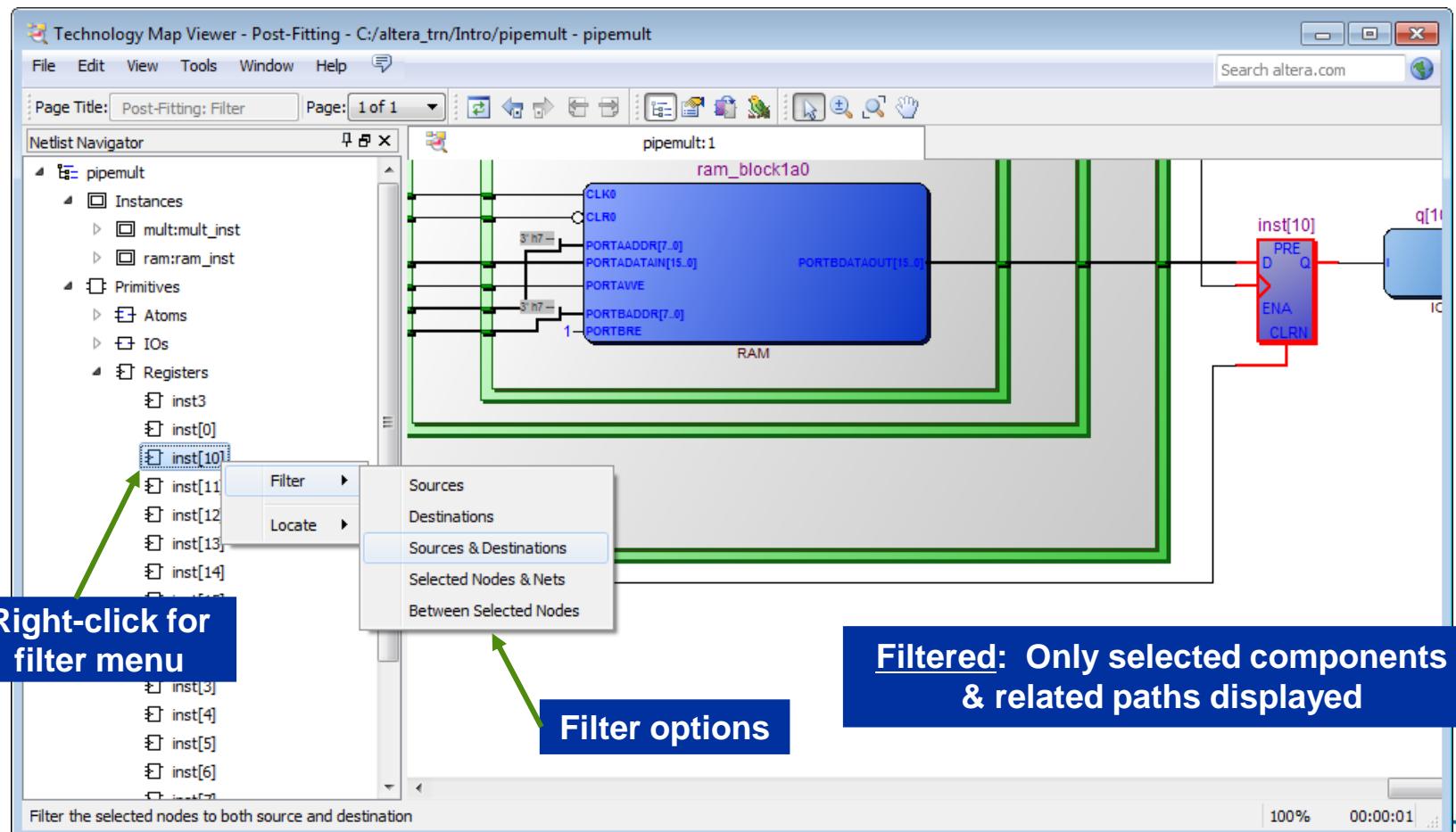
Using Hierarchy List

Expanding instances shows
• Instances
• Pins
• Nets



Filter Schematic

Unfiltered: All components & paths shown



Right-click for filter menu

Filter options

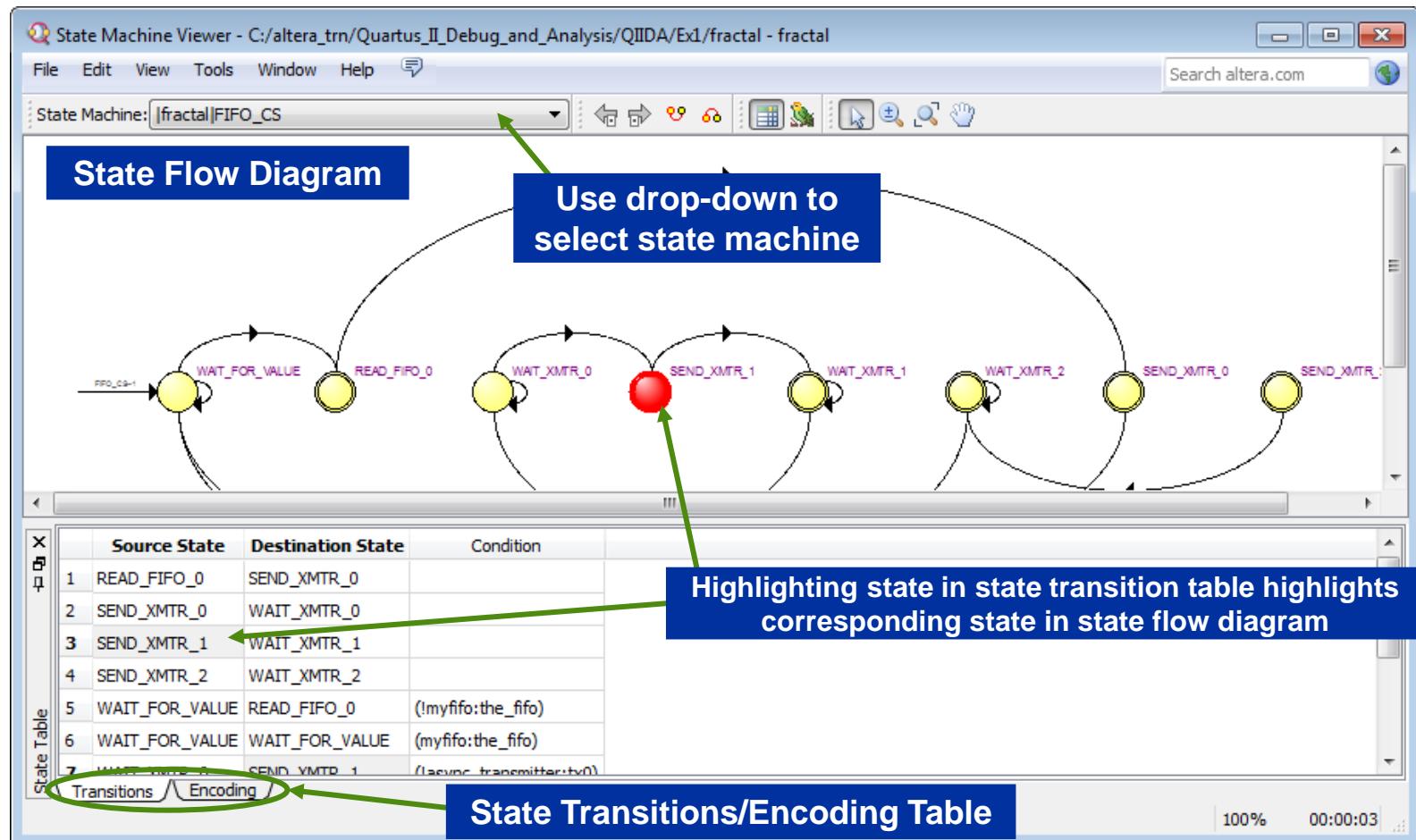
Filtered: Only selected components & related paths displayed

Other Features

- **Bird's Eye View:** overall panning view of design
- **Magnifying glass tool**
- **Page control**
- **Go to net driver:** traces net back to source driver
- **Lookup table (LUT) internal detail**
- **Cross-probing:** locate nodes from/to
 - Design files
 - Assignment Editor
 - Chip Planner
 - Resource Property Editor
 - RTL/Technology Map Viewers
 - Pin Planner
 - TimeQuest timing reports

State Machine Viewer

- Tools Menu → Netlist Viewers or Tasks window “Compile Design” tasks

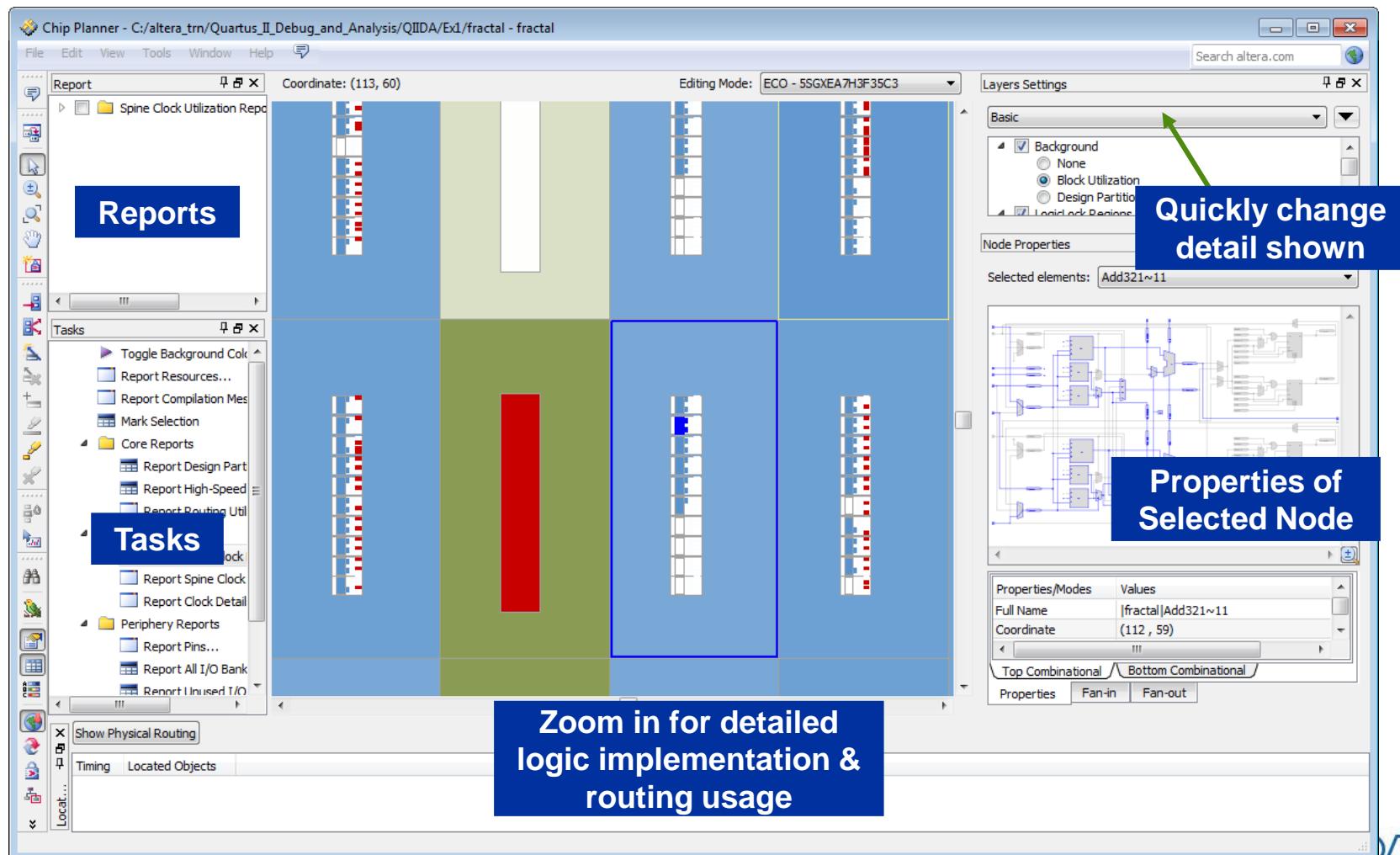


- **Graphical view of design resource usage in target device**
- **Displays**
 - Graphical layout of device resources
 - Routing channels between device resources
 - Global clock regions
- **Uses**
 - View placement of design logic
 - View connectivity between resources used in design
 - Make placement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

Chip Planner



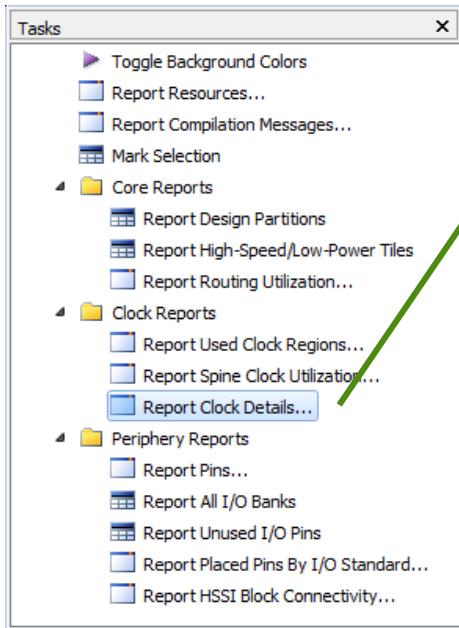
Tools Menu or
Tasks window “Compile Design” tasks



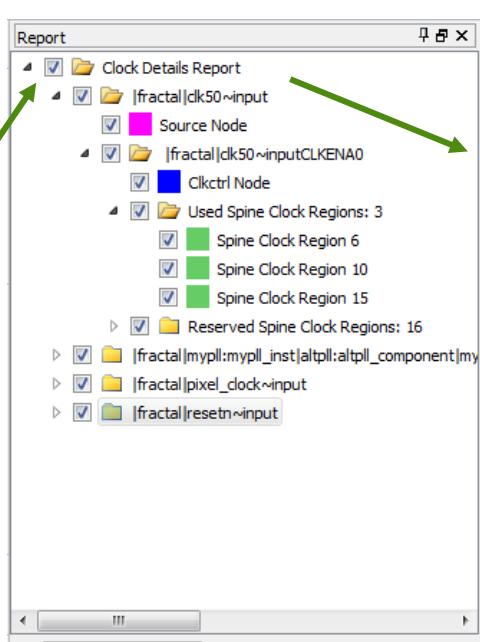
Chip Planner: Report & Task Windows

■ View→Task Window & View→Report Window

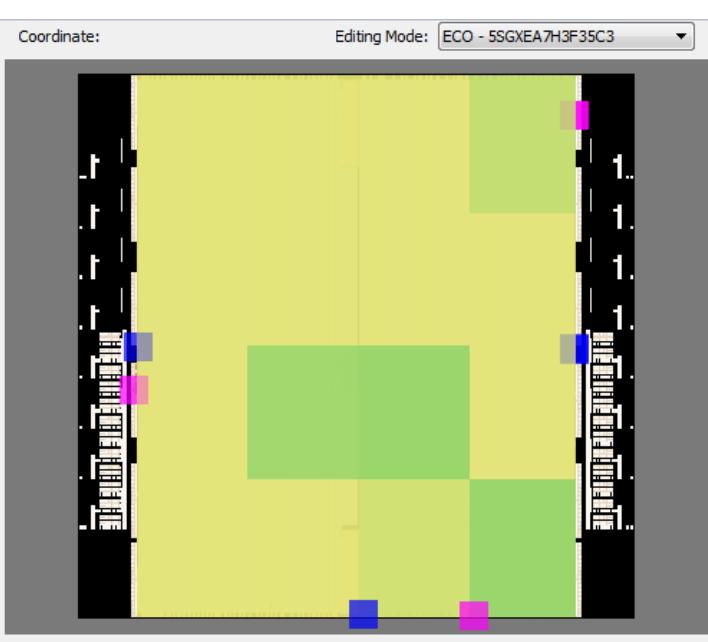
Run tasks that controls information shown



Reports generated for each task run



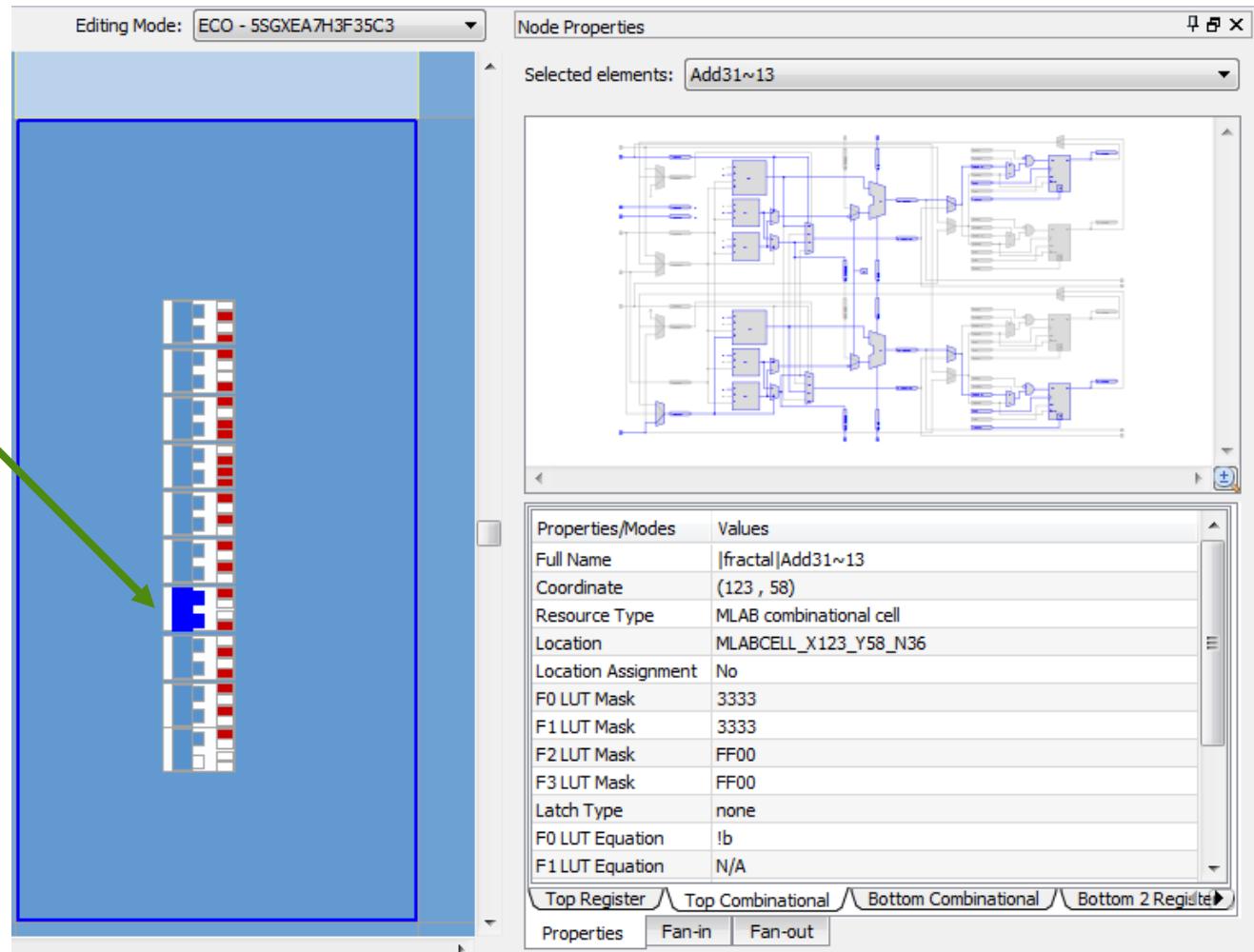
Enable report to see in floorplan view



Chip Planner: Show Block Resources

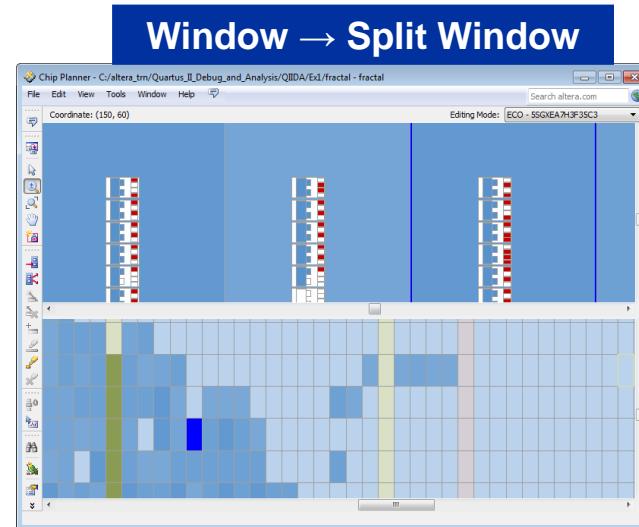
Click any block to view the internal resources detail

Double-clicking on the resource detail invokes the Resource Property Editor

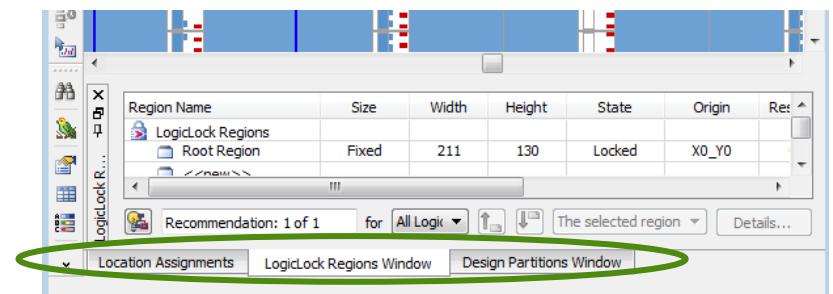


Splitter & Location Assignments

- **Split the floorplan into two views**
 - Allows for easier drag-and-drop operation
 - Can view at different zoom levels
- **Ability to perform assignments directly in Chip Planner and see the effect in the floorplan**



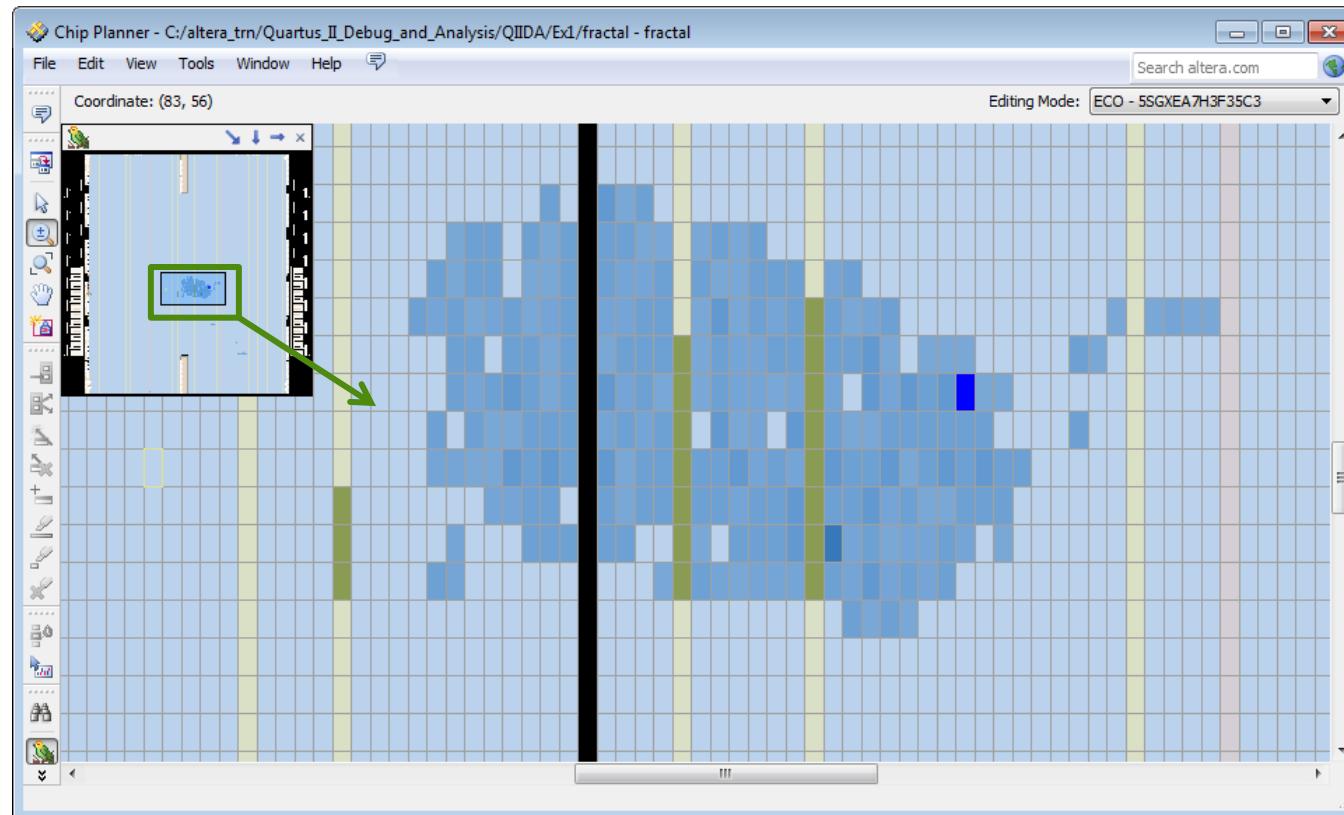
View → Location Assignment Window, LogicLock Regions Window, Design Partitions Window



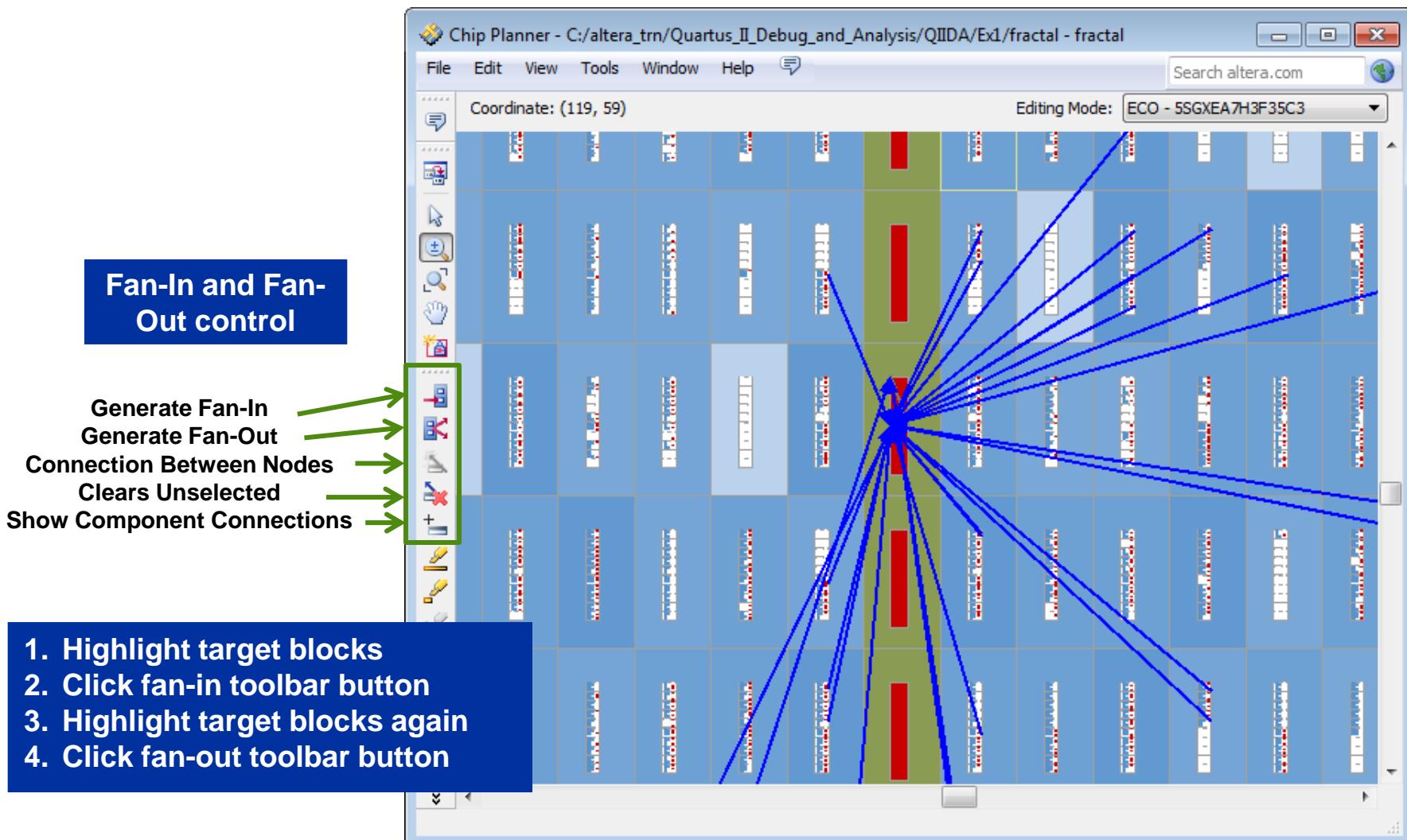
ALTERA
MEASURABLE ADVANTAGE™

Bird's Eye View

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

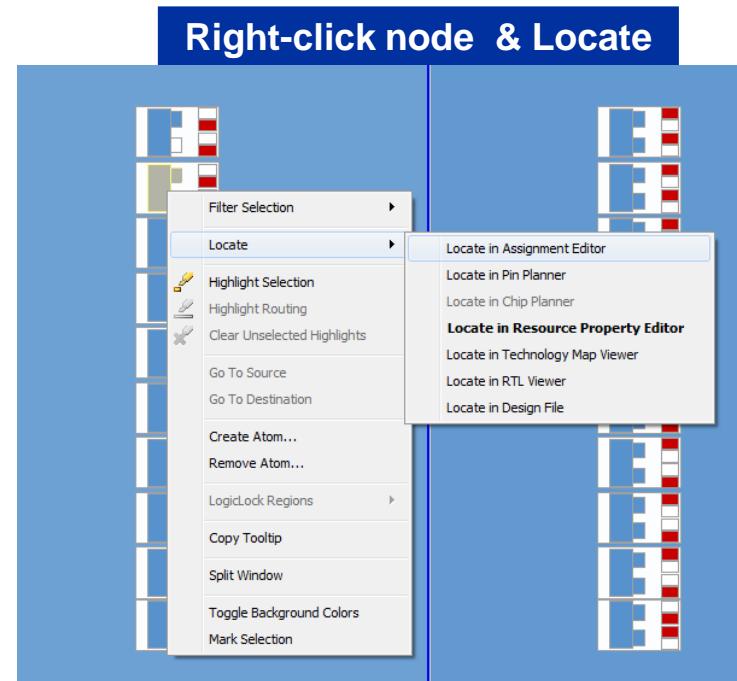


Displaying Fan-In & Fan-Out



Cross-Probing from/to Chip Planner

- Locate hierarchy blocks or specific logic from/to other Quartus II tools
- Project Navigator
- Compilation Report
- Design files
- RTL Viewer
- Technology Viewer
- Message window
- Pin Planner
- TimeQuest reports
- Resource Property Editor



Reasons for 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 Quartus II Software”](#)*

Test Your Knowledge: Compilation

1. What are the standard modules executed when performing a full compilation?
 - A.

2. What are some of the Quartus II software tools that can you use to understand how the design was processed?
 - A.

Exercise 3 Demonstration

*Demo should open automatically click
the link above if it doesn't*

Compilation Summary

- **Compilation includes synthesis & fitting**
- **Compilation Report contains detailed information on compilation results**
- **Use Quartus II software tools to understand how design was processed**
 - RTL Viewer
 - Technology Map Viewers
 - State Machine Viewer
 - Chip Planner

Compilation Support Resources

■ Quartus II Handbook chapters

- *Quartus II Incremental Compilation for Hierarchical & Team-Based Design* (Volume 1)
- *Best Practices for Incremental Compilation and Floorplan Assignments* (Volume 1)
- *Quartus II Integrated Synthesis* (Volume 1)
- *Analyzing Designs with Quartus II Netlist Viewers* (Volume 1)
- *Engineering Change Management with the Chip Planner* (Volume 2)

■ Training courses

- [Introduction to Incremental Compilation](#) (online)
- [Using the Quartus II Software: Chip Planner](#) (online)
- [Design Optimization Using Quartus II Incremental Compilation](#) (instructor-led)
- [Timing Closure with the Quartus II Software](#) (instructor-led)

Quartus II Software Design Series: Foundation

Settings & Assignments



© 2013 Altera Corporation—Public

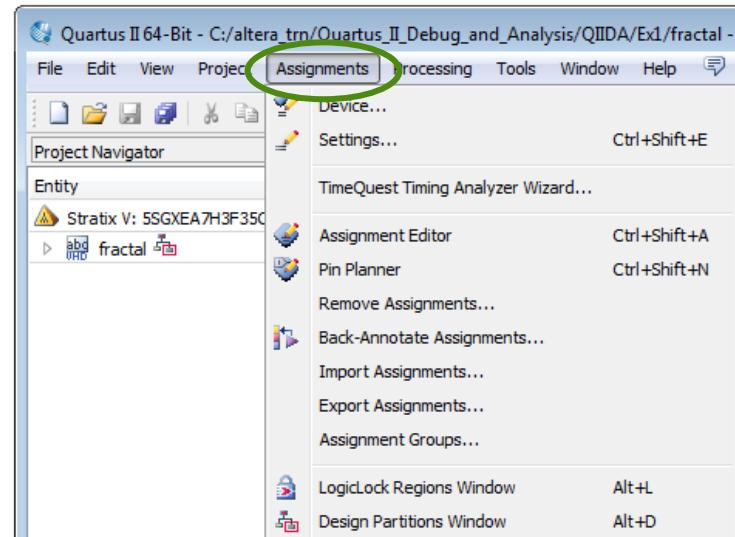
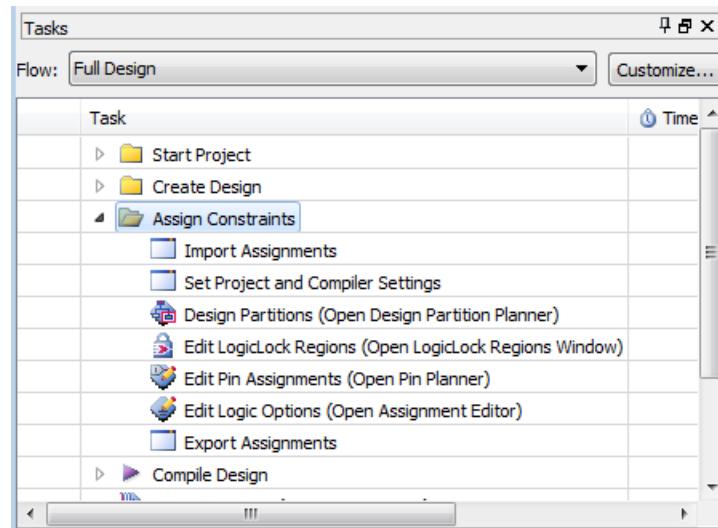


Setting & Assignments Objectives

- Define the difference between settings and assignments
- List some examples of setting assignments that can appear in a Quartus II project
- Create settings and assignments using various methods in the Quartus II 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

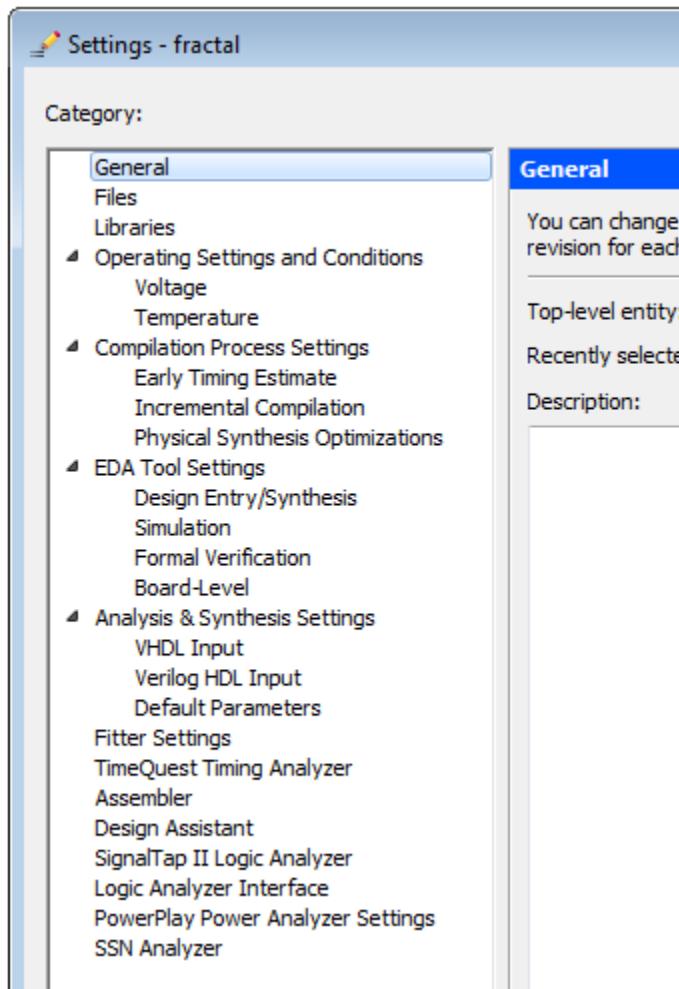


- **Project-wide switches that affect entire design**
- **Examples**
 - Device selection
 - Synthesis optimization
 - Fitter settings
 - Physical synthesis
 - Design Assistant
- **Located in Device and Settings dialog boxes**
 - **Assignments** menu
 - **Set Project and Compiler Settings** task in Tasks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

Settings Dialog Box

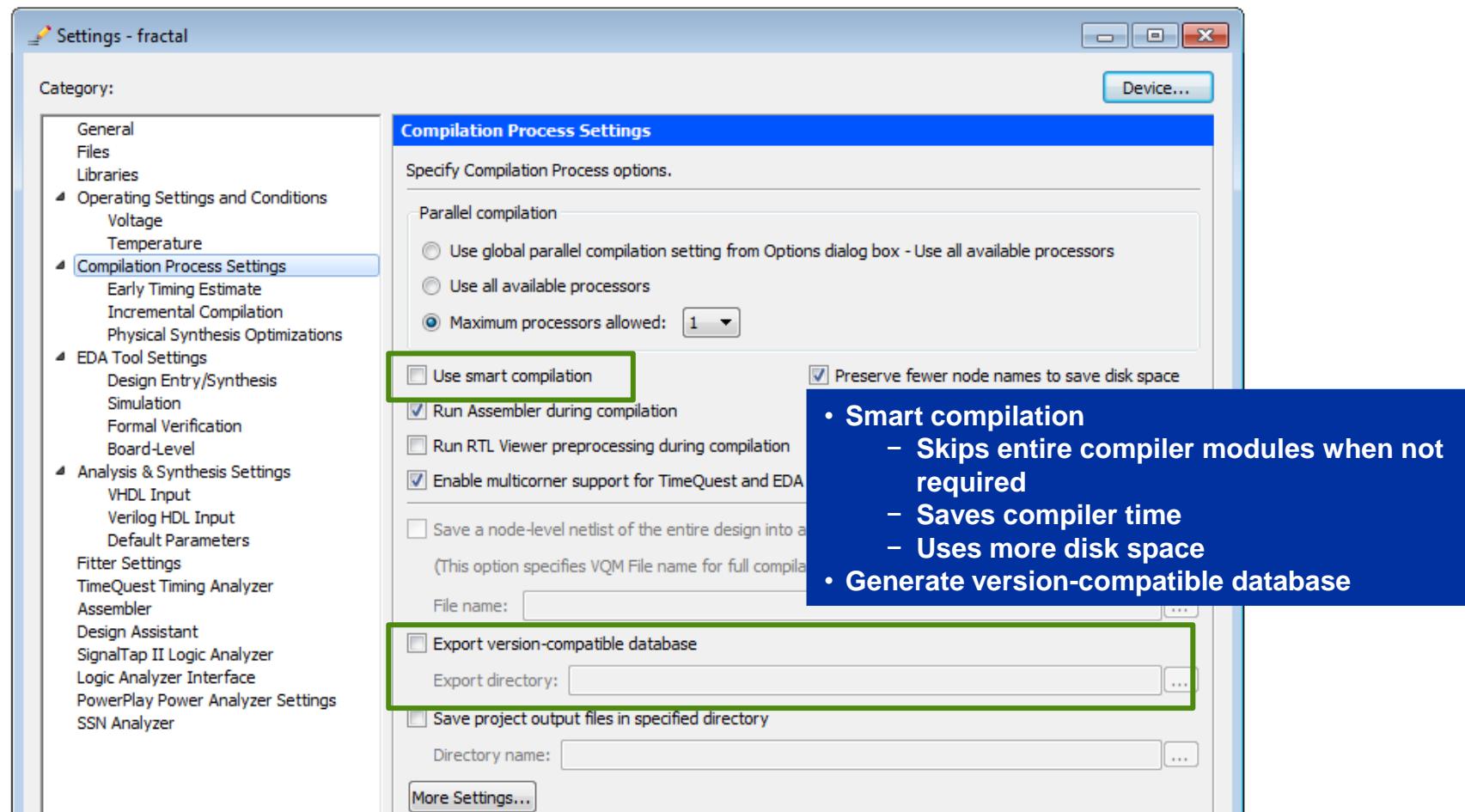
Change settings

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



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

Compilation Process Setting Examples



Tcl: set_global_assignment -name SMART_RECOMPILE ON

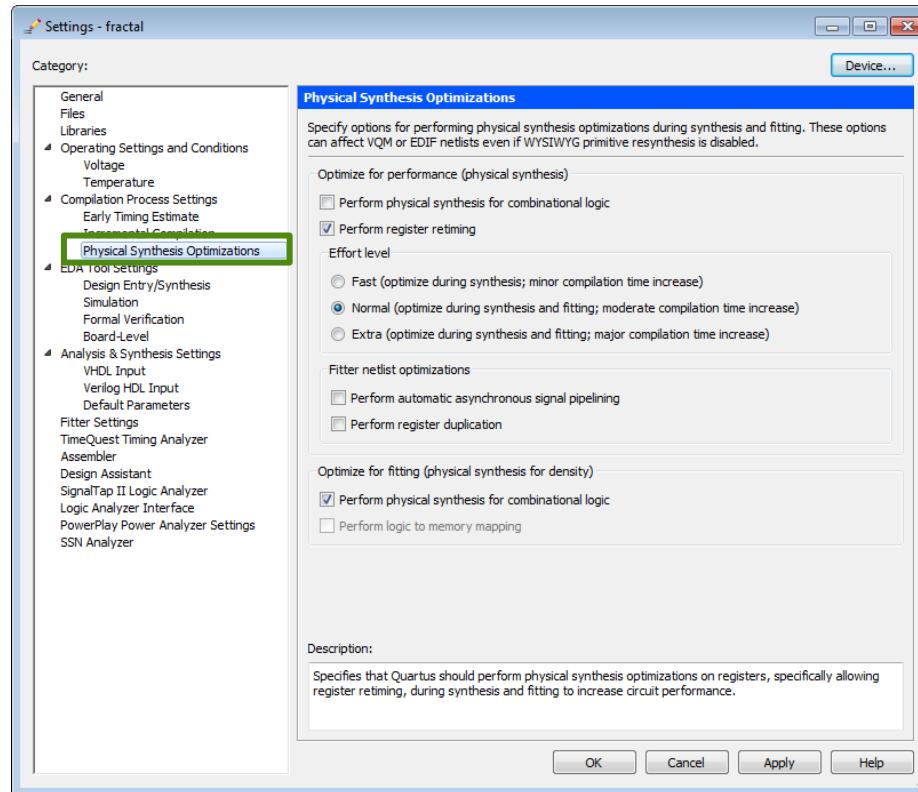
Version-Compatible Database

- **Migrating design between versions of Quartus II software**
- **Exports special database (export_db)**
- **Preserve compilation results between Quartus II software versions**
 - Re-run timing analysis or simulation with updated timing models
 - No need to fully recompile
- **Two methods to create**
 - Settings dialog box
 - Project menu (**Export Database**)

```
Tcl: set_global_assignment -name AUTO_EXPORT_VER_COMPATIBLE ON  
Tcl: set_global_assignment -name VER_COMPATIBLE_DB_DIR <directory_name>
```

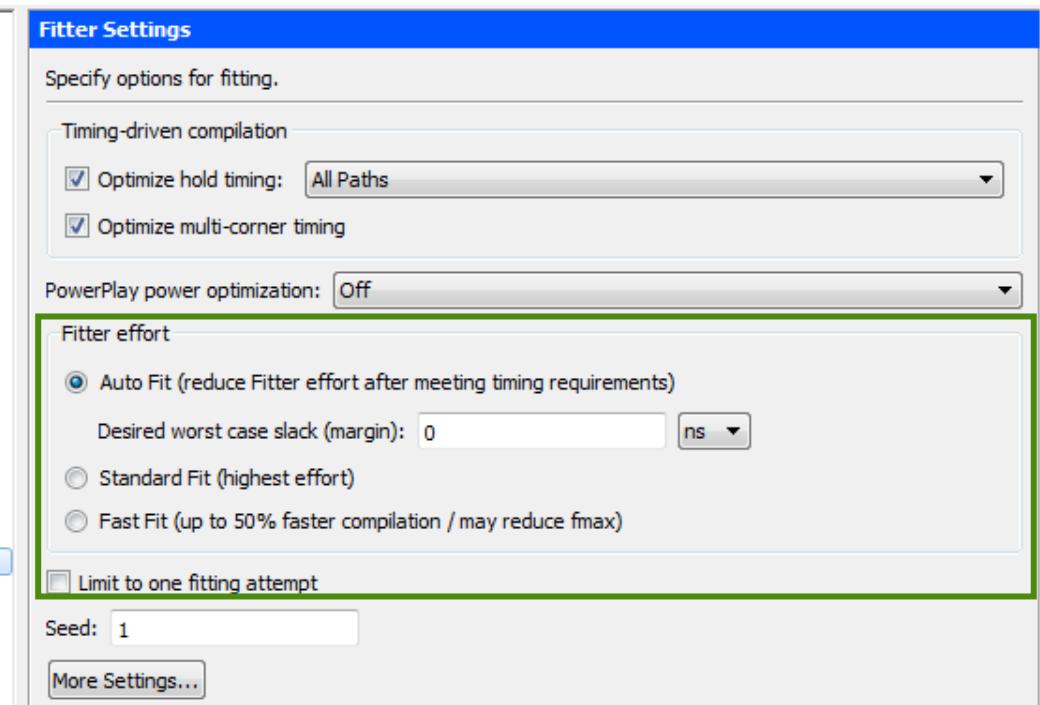
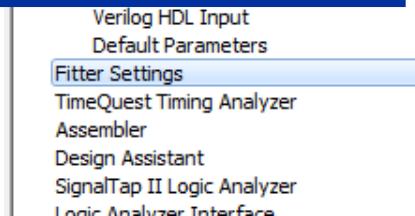
Physical Synthesis

- **Optimize during synthesis or re-synthesize based on Fitter output**
 - Makes incremental changes that improve results for a given placement
 - Compensates for routing delays from Fitter
 - Apply globally (Settings) or only to specific design entities (Assignment Editor)



Fitter Settings – Fitter Effort

- Standard Fit
 - Highest effort
 - Longest compile time
- Fast Fit
 - Faster compile
 - Possibly lesser performance
- Auto Fit
 - Compile stops after meeting timing
 - Conserves CPU time
 - Will mimic standard fit for hard-to-fit designs
 - Default for new designs
- One fitting attempt



Tcl: `set_global_assignment -name FITTER EFFORT "<Effort Level>"`

Assignments - Logic Options, Constraints

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

Assignment Editor

■ Provides spreadsheet assignment entry & display

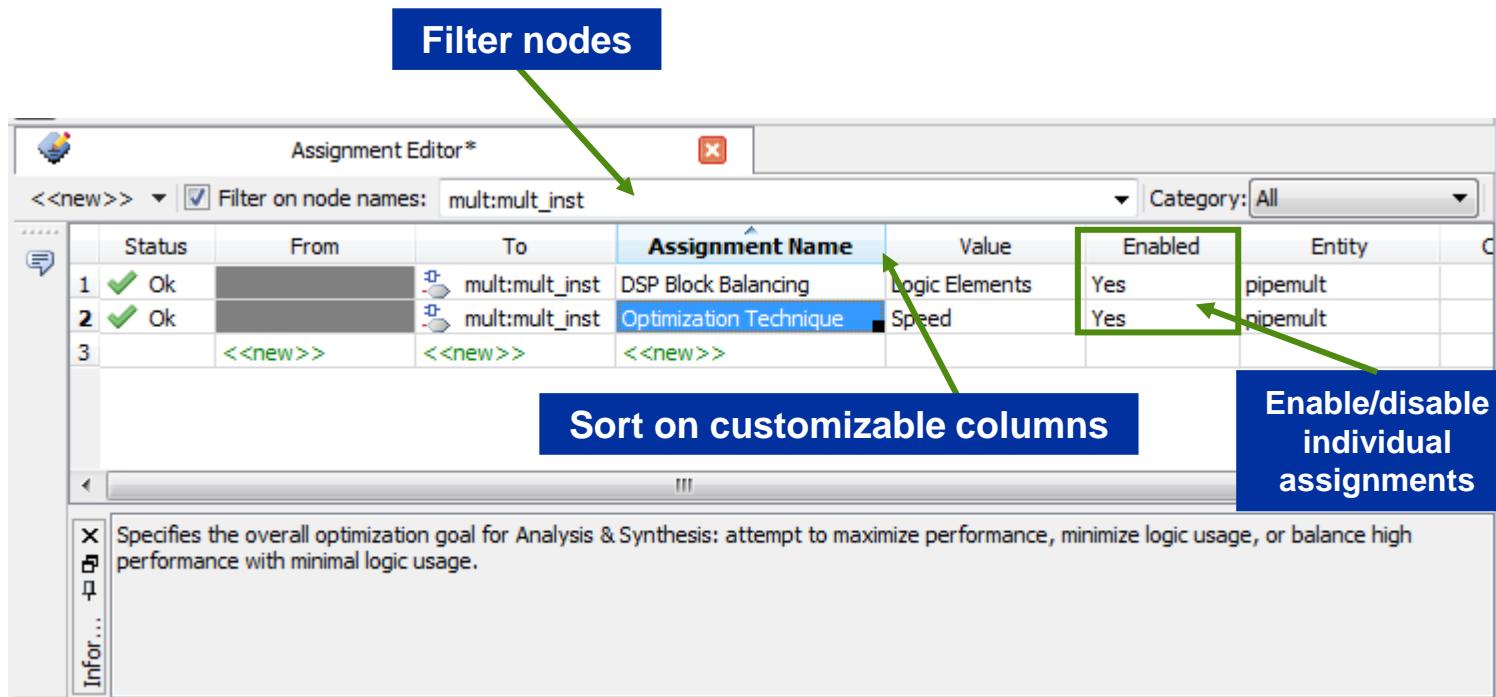
- Copy & paste support
- Multi-cell editing

Assignments menu or Tasks window

Filter nodes

Sort on customizable columns

Enable/disable individual assignments



Status	From	To	Assignment Name	Value	Enabled	Entity	Category
1 Ok		mult:mult_inst	DSP Block Balancing	Logic Elements	Yes	pipemult	
2 Ok		mult:mult_inst	Optimization Technique	Speed	Yes	pipemult	
3	<<new>>	<<new>>	<<new>>				

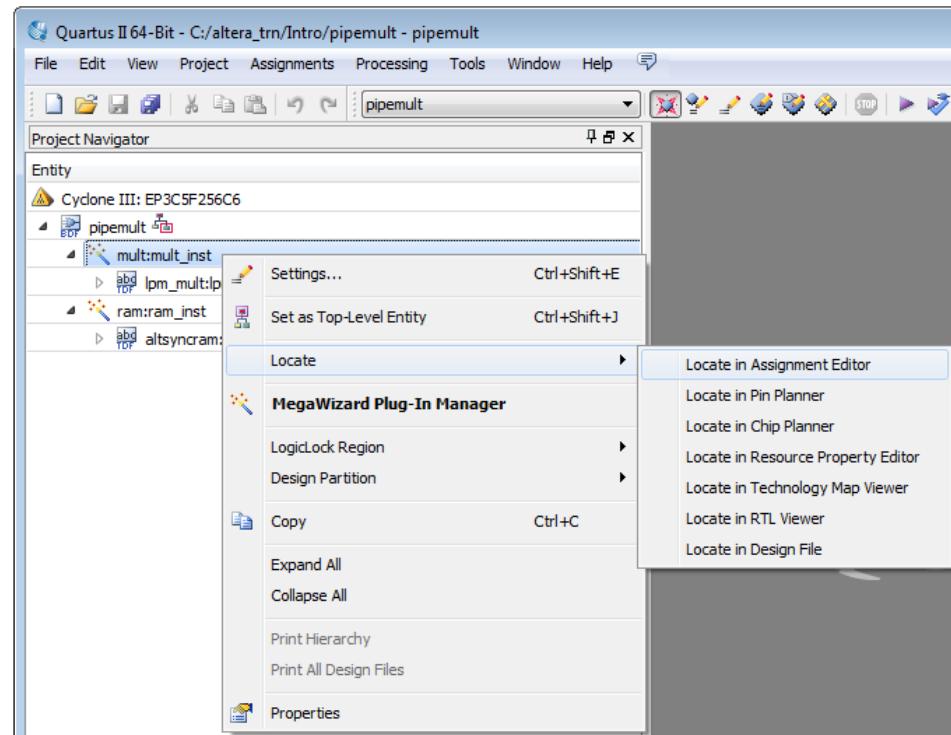
Specifies the overall optimization goal for Analysis & Synthesis: attempt to maximize performance, minimize logic usage, or balance high performance with minimal logic usage.

Creating Assignments: Cross-Probing

■ Cross-probe (Locate) to Assignment Editor

- Project Navigator
- Message window
- Compilation Report
- Design files

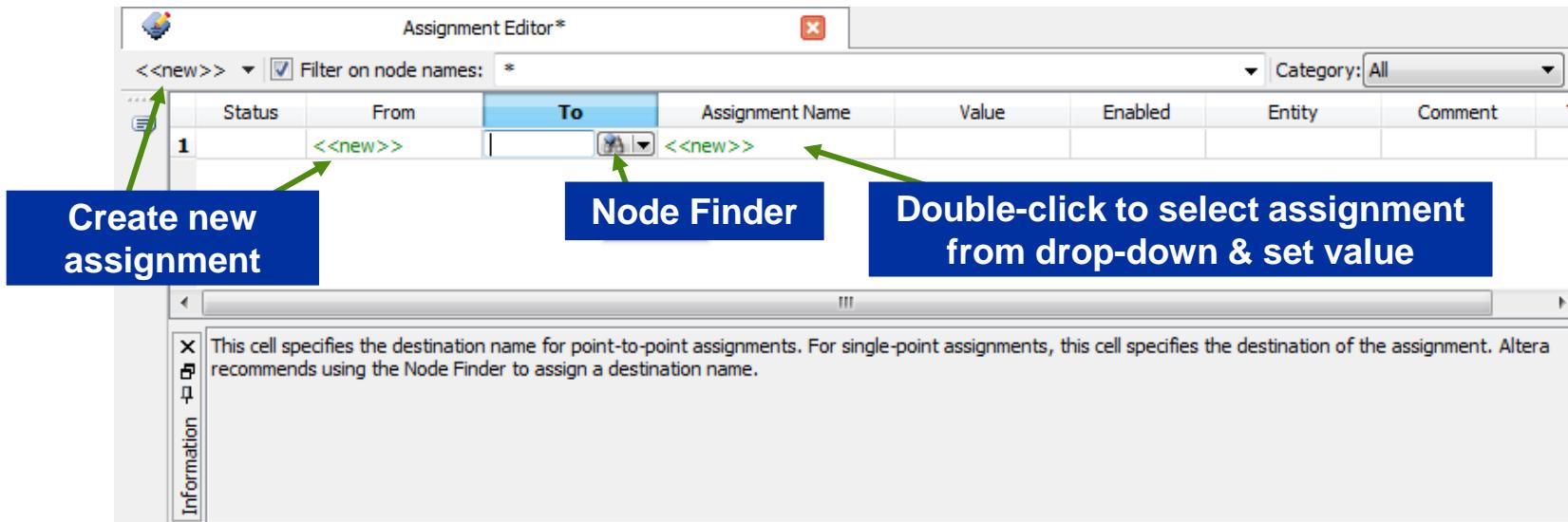
■ Assignment Editor Node Filter automatically filled in with cross-probed node(s)



■ Easiest method for creating assignments

Creating Assignments: Assignment Editor

- Locate from other tools
- Double-click on <<new>> button or <<new>> in the From or To columns
 - Type in object name
 - Click on Node Finder icon to search



Node Finder



Use filter to select the type of nodes to be selected

Locate nodes in a certain level of hierarchy

Search by name using wildcards (? Or *)

List of found nodes in selected entity & lower levels of hierarchy

using * or *)

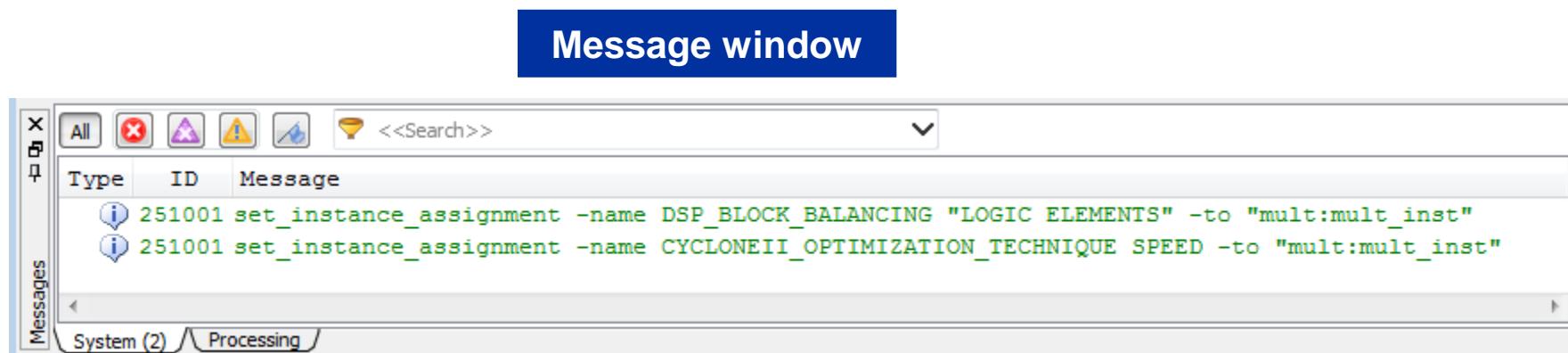
Select nodes on left & use arrows to move to the right

nodes in selected entity across all levels of hierarchy

Assignment Tcl Commands

■ Equivalent Tcl commands displayed as assignments are entered

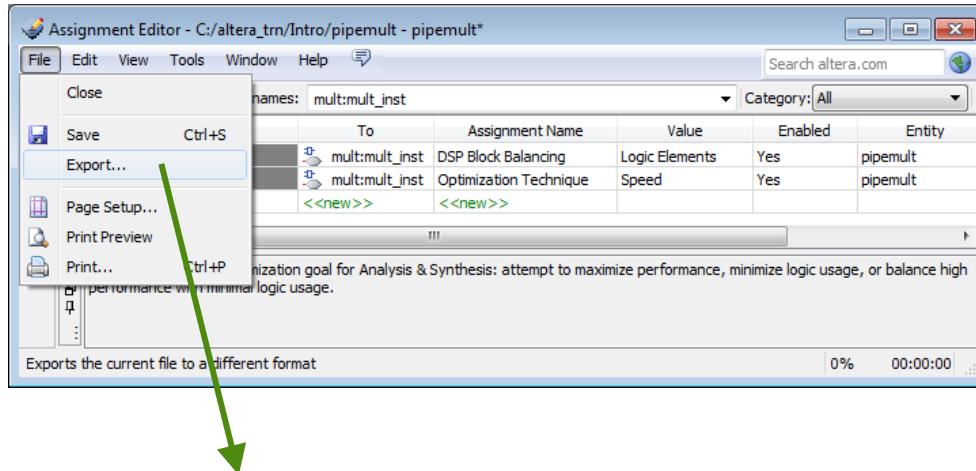
- Manually copy to create Tcl scripts
- Export command (**File** menu) writes all assignments to a Tcl file



Export CSV File Assignments (Excel)

■ Export to CSV file (File menu)

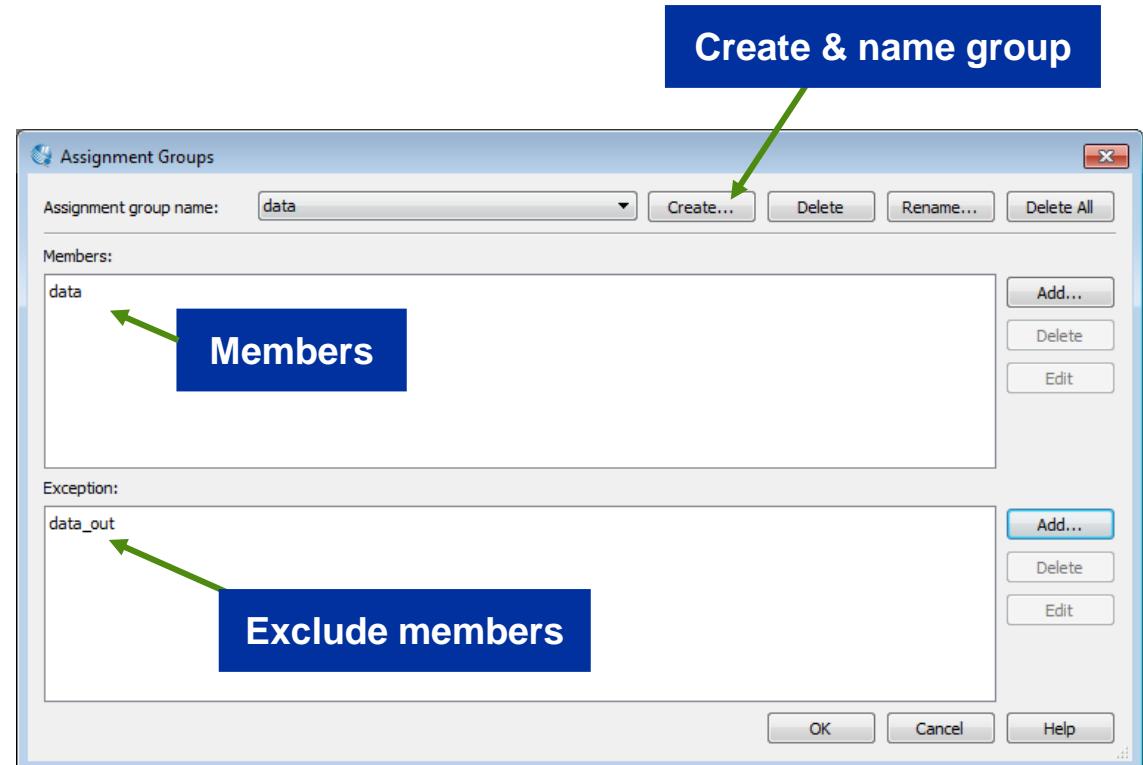
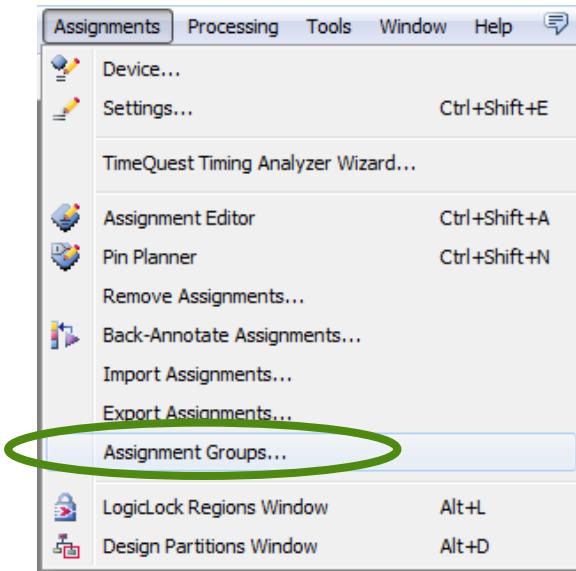
- Import and edit assignments in Excel



```
15 # Quartus II 64-Bit Version 12.1 Build 177 11/07/2012 SJ Full Version
16 # File: C:\altera_trn\Intro\pipemult_assignments.csv
17 # Generated on: Fri Dec 21 15:58:44 2012
18
19 Status From To Assignment Name Value Enabled Entity Comment Tag
20 Ok mult:mult_inst DSP Block Balancing Logic Elements Yes pipemult
21 Ok mult:mult_inst Optimization Technique Speed Yes pipemult
22 <>new><>new><>new>
23
```

Assignment Groups

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

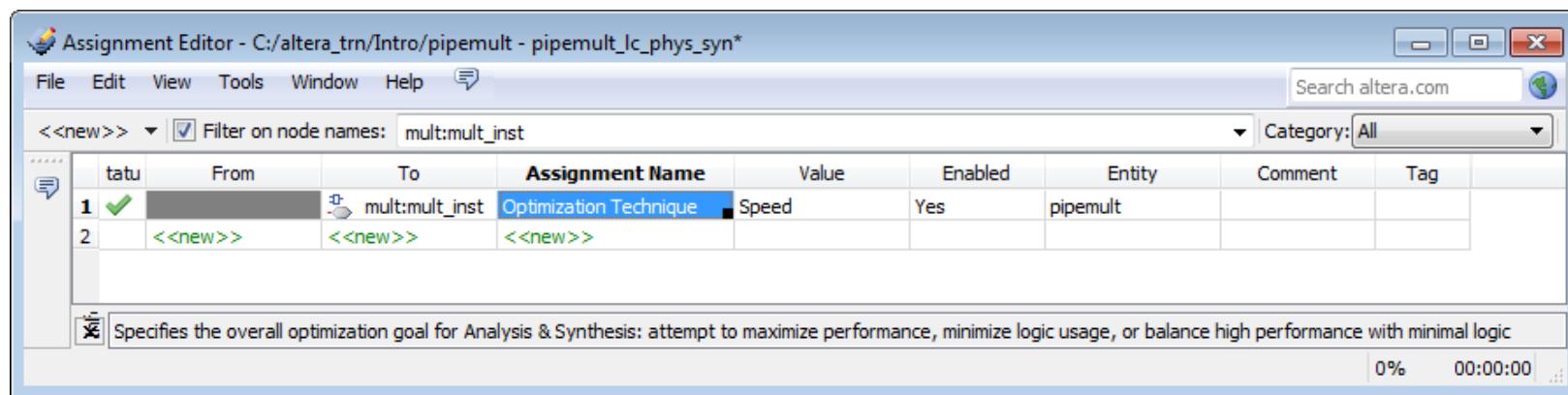


Example Assignments

- Optimization Technique
- PCI I/O

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 Analysis & Synthesis Settings
- Affects synthesis & logic mapping - applies to Quartus II synthesis



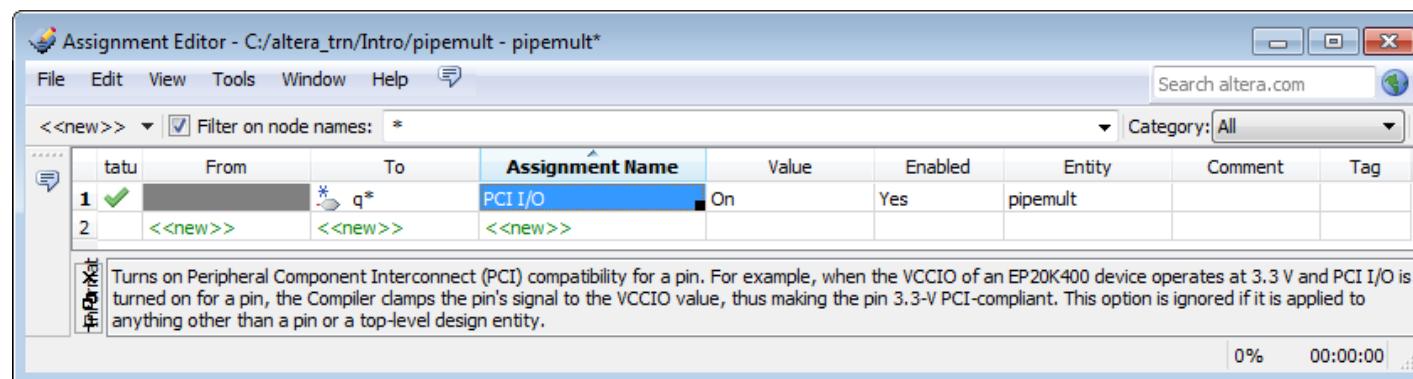
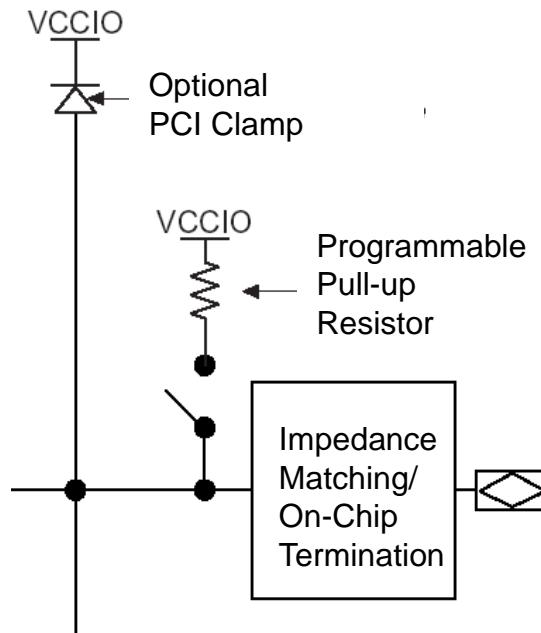
Tcl: `set_instance_assignment -name OPTIMIZATION_TECHNIQUE SPEED -to <node name>`

- Turns on PCI compatibility for pins

- Ignored if applied to anything other than a pin or a top-level design entity

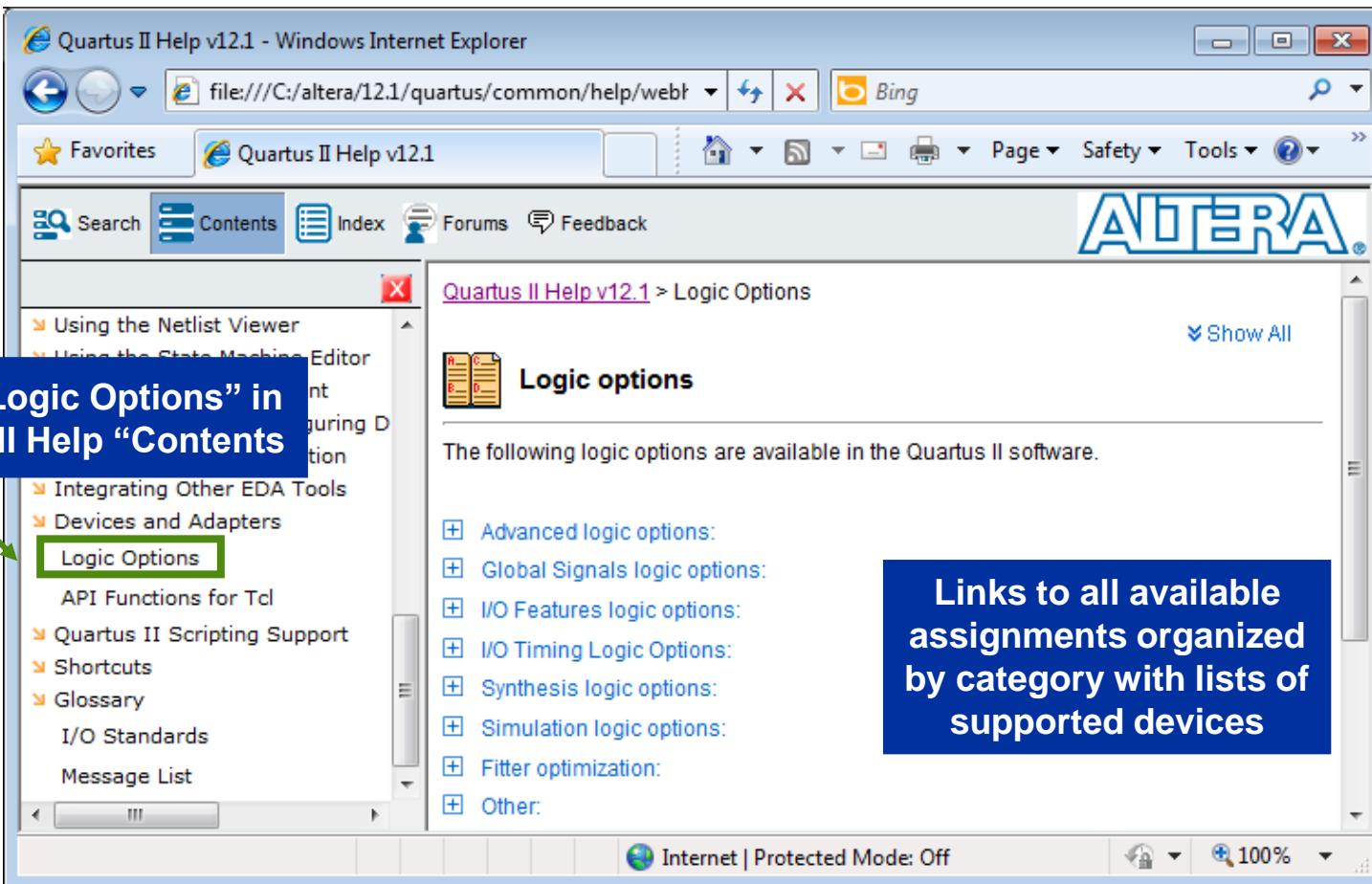
- Controls clamping diode located in the I/O elements

- For applicable families



Tcl: `set_instance_assignment -name PCI_IO ON -to <pin name>`

Available Logic Options (Assignments)



Quartus II Help v12.1 - Windows Internet Explorer

file:///C:/altera/12.1/quartus/common/help/webt

Bing

Favorites Quartus II Help v12.1

Search Contents Index Forums Feedback

ALTERA

Quartus II Help v12.1 > Logic Options

Logic options

The following logic options are available in the Quartus II software.

- ⊕ Advanced logic options:
- ⊕ Global Signals logic options:
- ⊕ I/O Features logic options:
- ⊕ I/O Timing Logic Options:
- ⊕ Synthesis logic options:
- ⊕ Simulation logic options:
- ⊕ Fitter optimization:
- ⊕ Other:

Select “Logic Options” in Quartus II Help “Contents”

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

Internet | Protected Mode: Off

100%

A green arrow points from the text "Select ‘Logic Options’ in Quartus II Help ‘Contents’" to the "Logic Options" link in the left sidebar.

A blue box highlights the "Logic Options" link in the left sidebar, and another blue box highlights the text "Links to all available assignments organized by category with lists of supported devices" on the right.

Updating QSF File

- **QSF not updated automatically when constraint entered or Assignment Editor saved**
- **QSF updated only when**
 - Project is saved (**File** menu)
 - Beginning of compilation
- **Change behavior to updating assignments immediately (Tools menu → Options → General → Processing)**
 - May impact software performance slightly due to file accesses

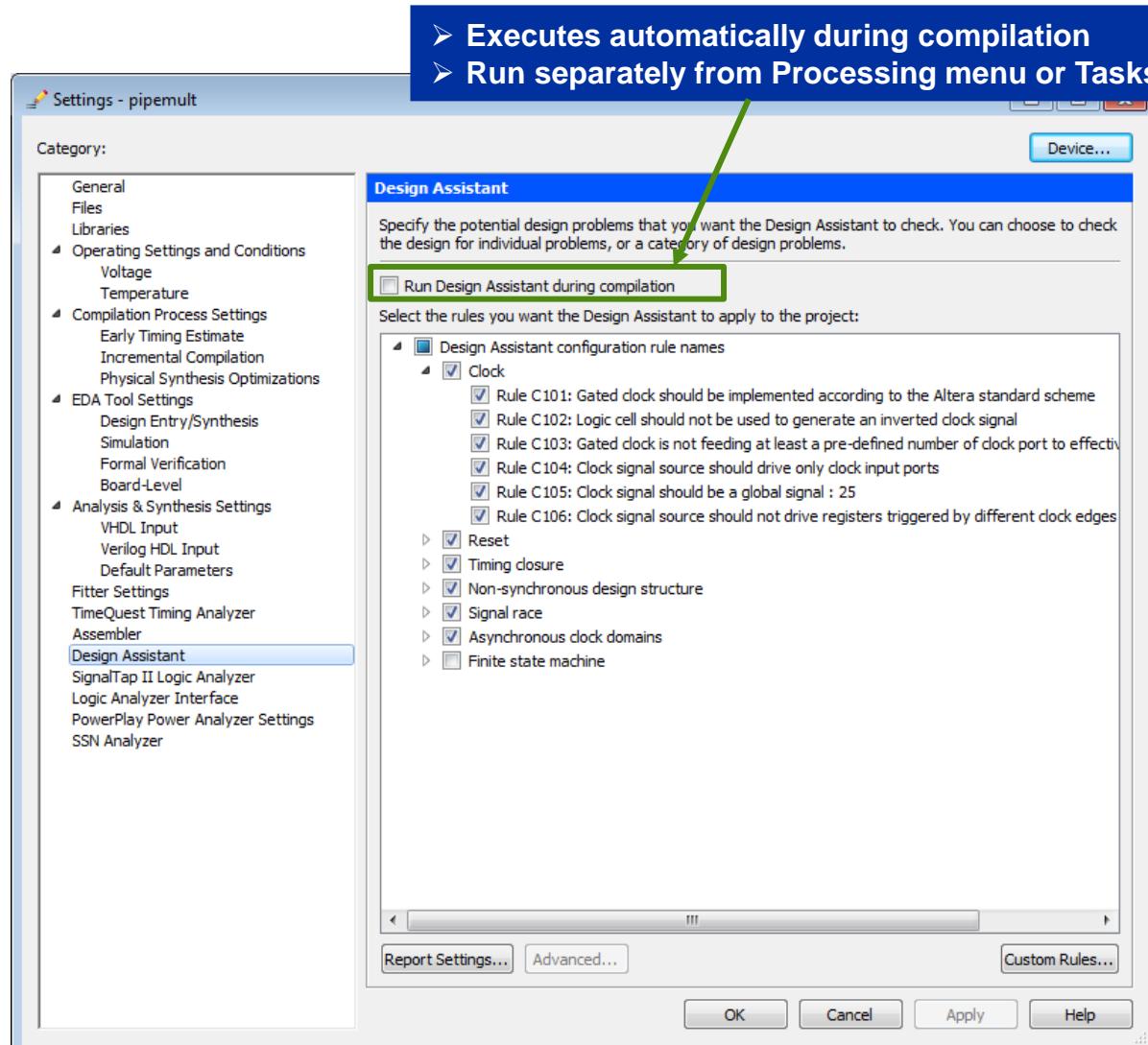
Design Assistance

■ Design Assistant

- Check design against sets of good design practice rules

■ Advisors

- Check current settings and assignments
- Make recommendations for different optimization goals

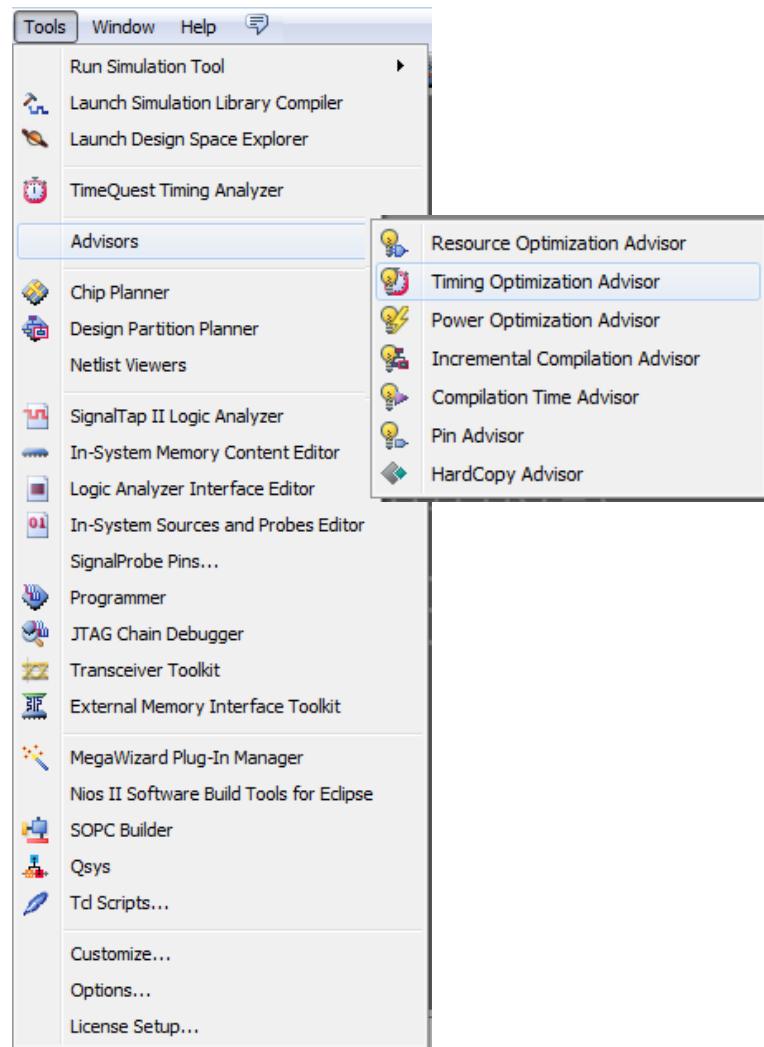


Potential design issues

- Clocks
- Reset
- Non-synchronous design structure
- Timing closure
- Asynchronous clock domain data transfers
- Signal race conditions
- FSM

Advisors

- **Provide design-specific recommendations (feedback) on optimizing designs**
- **Access through Tools menu or Tasks window**
- **Seven types**
 - Resource usage optimization
 - Timing (performance) optimization
 - Power optimization
 - Incremental compilation suggestions
 - Compilation time reduction
 - I/O-related recommendations
 - HardCopy flow recommendations
- ***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. The 'Maximum Frequency (fmax)' category is expanded, showing a sub-section for 'Use High-Effort fmax Optimization Settings' which is also expanded. This section contains several recommendations, each with a yellow warning icon and a link. Some recommendations have a green checkmark next to them. On the right, a detailed view of the 'Use High-Effort fmax Optimization Settings' section is shown. It has a blue header bar. Below it, a section titled 'Use High-Effort fmax Optimization Settings' contains a list of recommendations. One recommendation, 'Enable Beneficial Skew Optimization', has a green checkmark and is highlighted with a blue box and an arrow pointing to it. A 'Correct All Settings' button is at the bottom. A blue box at the bottom left of the main window contains the text: 'Green checkmark indicates settings already in use'.

Timing Optimization Advisor

Timing Summary

- How to use the Timing Optimization Advisor
- General Recommendations
 - Get more information
 - Create a revision
 - Use smart compilation
 - Use the Design Assistant
 - Review timing constraints - TQ
 - Select a Faster Speed Grade Device
- Maximum Frequency (fmax)
 - Use High-Effort fmax Optimization Settings
 - 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 fast output enable...
 - Use higher drive strength
- Hold Time & Minimum Delay Timing
 - Use multi-corner timing
 - Turn on Optimize Hold Timing
 - Turn off Auto Packed Registers
 - Enable clocking topology analysis during routing
- Metastability Optimization
 - Optimize metastability during Fitting
 - Increase the Length of Synchronizers to Protect

Use High-Effort fmax Optimization Settings

Use High-Effort fmax Optimization Settings

The following recommendations are provided to improve the maximum frequency (fmax) of your design.

- Optimize for speed
- Use physical synthesis optimizations
- Enable Logic Cell Insertion - Logic Duplication
- Set maximum router effort
- Avoid unrelated logic register packing
- Enable Beneficial Skew Optimization

Use the "Correct All Settings" button to implement all recommendations listed above.

Correct All Settings

Green checkmark indicates settings already in use

Adjust settings

Test Your Knowledge: Compilation

1. What is the difference between settings and assignments?

A.

2. Is the optimization technique a setting or an assignment?

A.

3. Is the target device a setting or an assignment?

A.

Exercise 4 Demonstration

*Demo should open automatically click
the link above if it doesn't*

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**
- **Design Assistant & Optimization Advisors help improve design results through design rule checking and settings recommendations**

■ Quartus II Handbook chapters (all Volume 2)

- *Area & Timing Optimization*
- *Power Optimization*
- *Netlist Optimizations & Physical Synthesis*

■ Training courses

- [Timing Closure with the Quartus II Software](#) (instructor-led)
- [Timing Closure Using Quartus Advisors and Design Space Explorer](#) (online)
- [Timing Closure Using Quartus II Physical Synthesis Optimizations](#) (online)

Quartus II Software Design Series: Foundation

I/O Planning



© 2013 Altera Corporation—Public



- **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**

I/O Planning Section Objectives

- **Assign pin locations and other properties using Pin Planner**
- **Back annotate fitter chosen pin assignments**
- **Verify that IO assignments are valid**

- **Creating device I/O location and related assignments**
- **Analysis**
 - Compiler process
 - Live I/O checking

Creating I/O-Related Assignments

- Pin Planner
- Import from spreadsheet in CSV format
- Type directly into QSF file
- Scripting

Note: Other methods/tools are available in the Quartus II software to make I/O assignments. The above are the most common or recommended.

- **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 or
“Assign Constraints”
folder in Tasks window**

Pin Planner Window

The screenshot shows the Pin Planner window with several labeled panes:

- Groups list**: A table showing pin groups, with an entry for "rdaddress[4..0]" as an Input Group.
- Report Pane**: A tree view of reports, including "Pin Resource Reports" and "Pin Characteristic Reports".
- Tasks Pane**: A list of tasks under "Early Pin Planning" and "Change View".
- All Pins list**: A table listing pins with columns for Node Name, Direction, Location, I/O Bank, VREF Group, I/O Standard, Reserved, and Current Strength. The table shows several input pins (e.g., "dk1", "dataa[7]", "dataa[6]", "dataa[5]", "dataa[4]") assigned to IOBANK_7A.
- Package View**: A grid representing the physical package layout for a Cyclone V device (5CEBA9F23C7). The grid shows pins numbered 1 to 22 and labeled with letters A through Z. A specific pin, "dataa[4]", is highlighted in red.

A green arrow points from the "Toolbar" label to the toolbar area on the left side of the window.

Pin Planner Window Panes

■ 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

■ Groups list

- Similar to All Pins list displaying only groups & buses
- Make bus and group assignments
- Create new user-defined groups

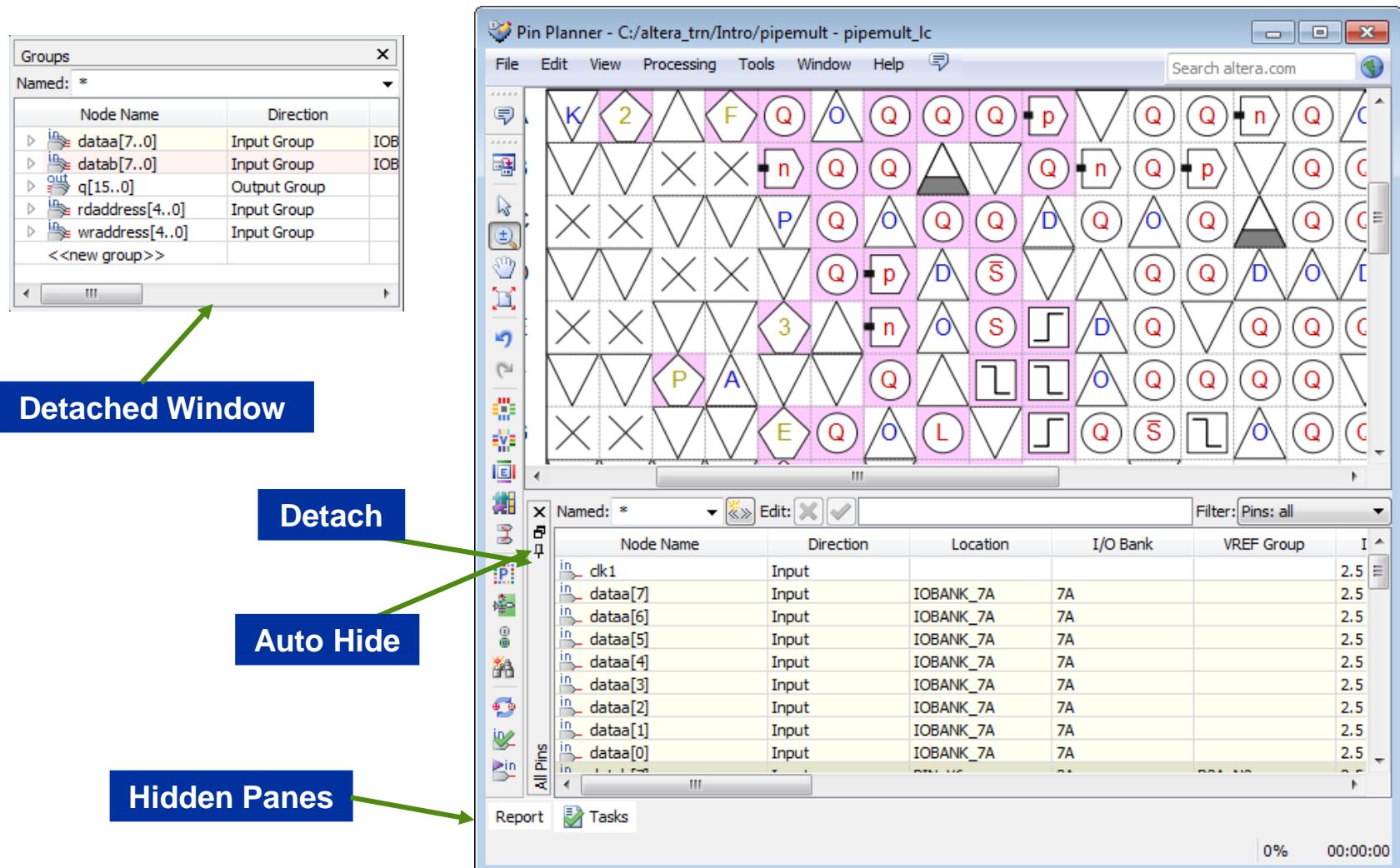
■ Tasks Pane

- Perform tasks such as Early Pin Planning and view changes

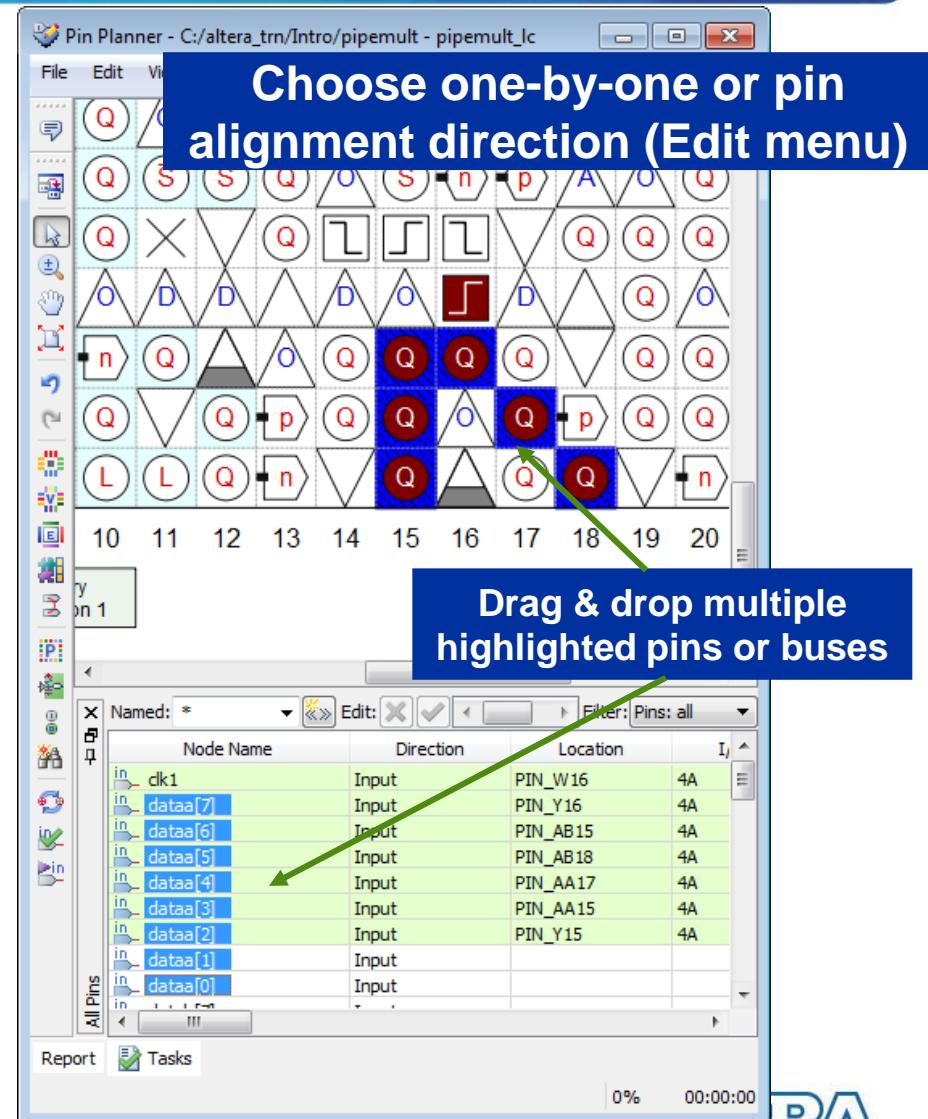
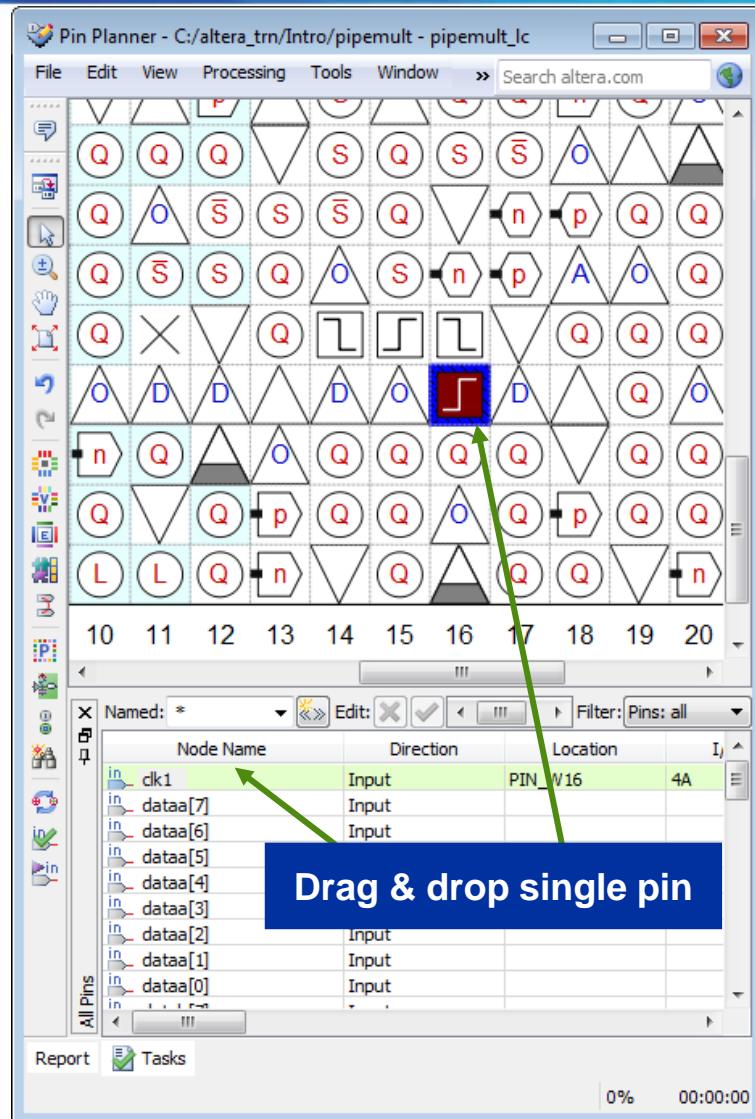
■ Report Pane

- Quickly enable/disable views generated by some view change tasks

Dockable Hideable Window Panes



Assigning Pin Locations Using Pin Planner



Assigning Pin Locations Using Pin Planner (2)

The screenshot shows the Pin Planner interface for an Altera device. The main workspace displays a grid of pins (numbered 3 to 19) and I/O banks. A callout box with a green arrow points to the top row of pins, with the text: "Drag & drop to I/O bank, VREF block or device edge". Another callout box with a green arrow points to the "Pin Properties" panel on the right, with the text: "Click pin or I/O bank to display pin properties". The "Pin Properties" panel shows details for PIN_B17, including its I/O bank (7A), VREF group (B7A_N0), and edge (TOP). The "Filter" dropdown in the bottom right of the main window is set to "Pins: all".

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

Click pin or I/O bank to display pin properties

Pin Properties

Pin number	PIN_B17
Node name:	
I/O Standard:	
Reserved:	
Properties:	
Name	Value
I/O bank	7A
VREF group	B7A_N0
Edge	TOP
General function	Column I/O
Special function	DIFFIO_TX_T14n DIFFOUT_T14n DQ2T
Pad ID	462
Pad group	7

Filter: Pins: all

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate	Differential
in_dk1	Input	PIN_W16	4A	B4A_N0	2.5 V (default)		12mA (default)		
in_dataa[7]	Input	IOBANK_7A	7A		2.5 V (default)		12mA (default)		
in_dataa[6]	Input	IOBANK_7A	7A		2.5 V (default)		12mA (default)		
in_dataa[5]	Input	IOBANK_7A	7A		2.5 V (default)		12mA (default)		
in_dataa[4]	Input	IOBANK_7A	7A		2.5 V (default)	1			
in_dataa[3]	Input	IOBANK_7A	7A		2.5 V (default)	1			
in_dataa[2]	Input	IOBANK_7A	7A		2.5 V (default)	1			
in_dataa[1]	Input	IOBANK_7A	7A		2.5 V (default)	1	12mA (default)		
in_dataa[0]	Input	IOBANK_7A	7A		2.5 V (default)	1	12mA (default)		

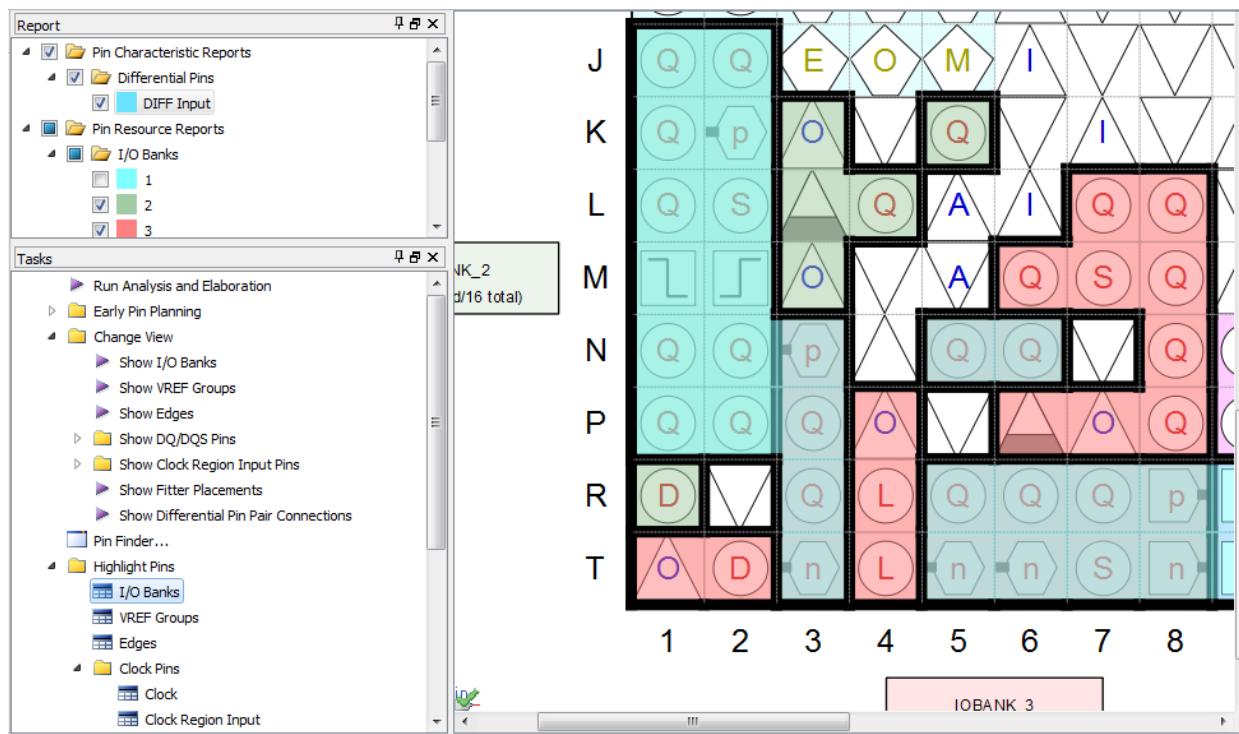
Assigning Pin Locations Using Pin Planner (3)

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

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate	Differential
clk1	Input	PIN_W16	4A	B4A_N0	2.5 V (default)		12mA (default)		
dataa[7]	Input	PIN AA13							
dataa[6]	Input	PIN_AA20	IOBANK_4A	Column I/O	DIFFIO_TX_B61p, DIFFOUT_B61p, DQ8B				
dataa[5]	Input	PIN_AA22	IOBANK_4A	Column I/O	DIFFIO_TX_B64p, DIFFOUT_B64p, DQ8B				
dataa[4]	Input	PIN_AB5	IOBANK_3B	Column I/O	DIFFIO_TX_B33p, DIFFOUT_B33p, DQ5B				
dataa[3]	Input	PIN_AB6	IOBANK_3B	Column I/O	DIFFIO_TX_B33n, DIFFOUT_B33n				
dataa[2]	Input	PIN_AB7	IOBANK_3B	Column I/O	DIFFIO_TX_B36p, DIFFOUT_B36p				
dataa[1]	Input	PIN_AB8	IOBANK_3B	Column I/O	DIFFIO_TX_B37p, DIFFOUT_B37p, DQ5B				
dataa[0]	Input	PIN_AB10	IOBANK_3B	Column I/O	FPLL_BL_CLKOUT1, FPLL_BL_CLKOUTn, DIFFIO_TX_B45n, DIFFOUT_B45n, DQ6B				
		PIN_AB11	IOBANK_3B	Column I/O	FPLL_BL_CLKOUT0, FPLL_BL_CLKOUTp, FPLL_BL_FB, DIFFIO_TX_B45p, DIFFOUT_B45p, DQ6B				
		PIN_AB12	IOBANK_4A	Column I/O	DIFFIO_TX_B49p, DIFFOUT_B49p, DQ7B				
		PIN_AB13	IOBANK_4A	Column I/O	RZQ_0, DIFFIO_TX_B49n, DIFFOUT_B49n				

Pin Planner Tasks & Report Windows

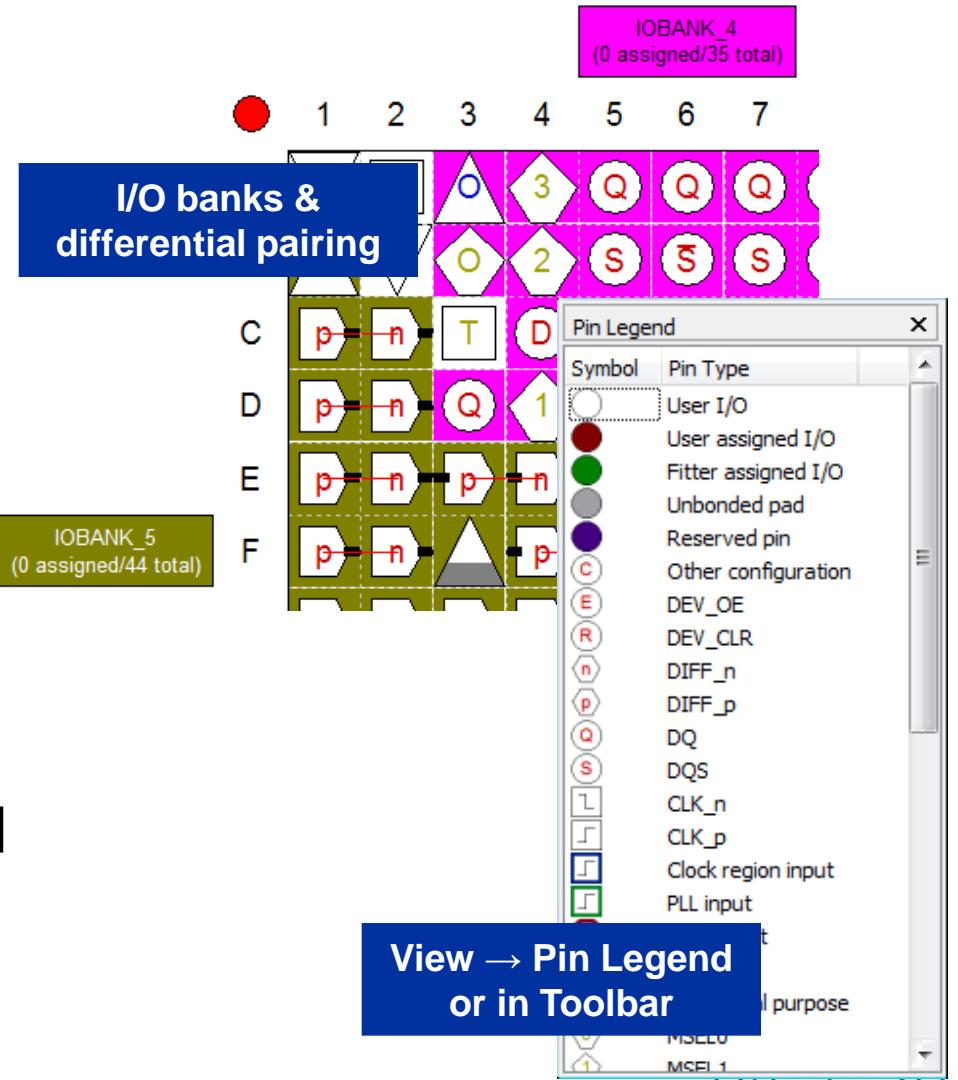
- Use Tasks window to access common tasks
- Highlight Pins tasks will generate overlays listed in the Report window



Other Pin Planner Features

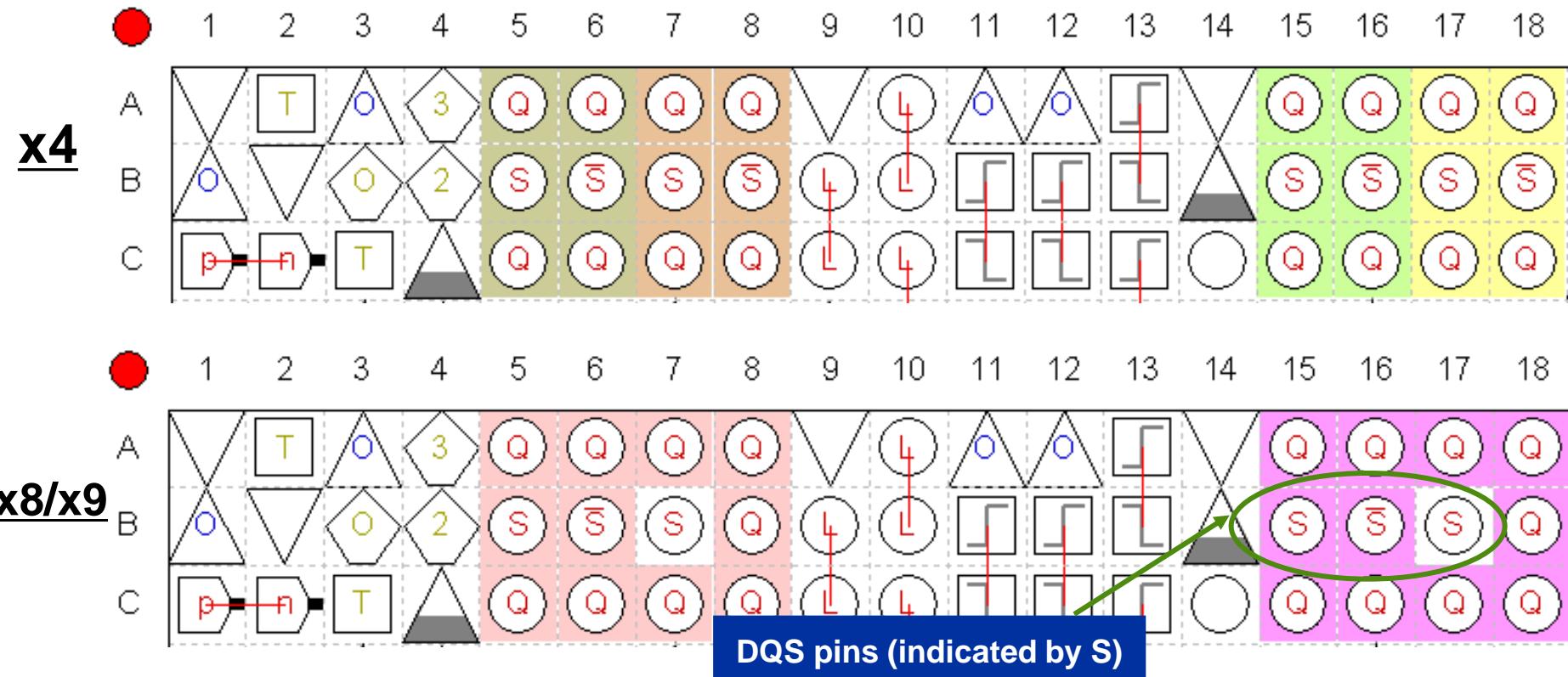
- **Displays (View ⇒ Show, Toolbar buttons, or right-click in Package View)**

- Device edges
- I/O banks
- VREF groups
- Differential pin pairing
- DQ/DQS pins



Additional 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



Pin Migration View

- Select migration devices in Device Settings
- View & compare pin function differences between migration devices
- Package View adjusts to prevent non-migratable assignments

View → Pin Migration View or right-click in Package View

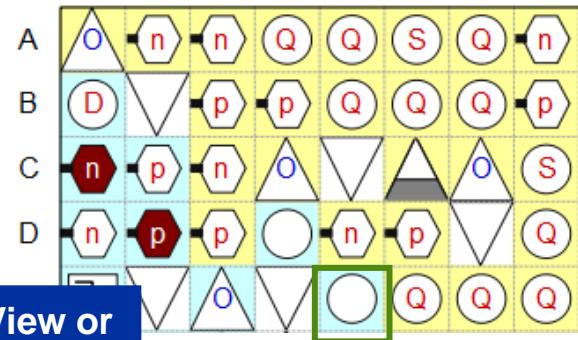
Pin Migration View

Current Device: EP4CE6F17C6

Pin Number	Migration Result			Migration Devices					
	Pin Function	I/O Bank	VREF Group	EP4CE6F17C6			EP4CE22F17C6		
				Pin Function	I/O Bank	VREF Group	Pin Function	I/O Bank	VREF Group
PIN_F6	GND			Column I/O	8	B8_N0	GND		
PIN_E5	VCCA3			Row I/O	1	B1_N0	VCCA3		
PIN_E5	GNDA3			Row I/O	1	B1_N0	GNDA3		
PIN_D4	VCCD_PLL3			Row I/O	1	B1_N0	VCCD_PLL3		
PIN_T8	Column I/O	3	B3_N0	Column I/O	3	B3_N0	Dedicated Clock	3	B3_N0
PIN_R8	Column I/O	3	B3_N0	Column I/O	3	B3_N0	Dedicated Clock	3	B3_N0
PIN_T9	Column I/O	4	B4_N0	Column I/O	4	B4_N0	Dedicated Clock	4	B4_N0
PIN_R9	Column I/O	4	B4_N0	Column I/O	4	B4_N0	Dedicated Clock	4	B4_N0
PIN_B9	Column I/O	7	B7_N0	Column I/O	7	B7_N0	Dedicated Clock	7	B7_N0
PIN_A9	Column I/O	7	B7_N0	Column I/O	7	B7_N0	Dedicated Clock	7	B7_N0
PIN_B8	Column I/O	8	B8_N0	Column I/O	8	B8_N0	Dedicated Clock	8	B8_N0
PIN_A8	Column I/O	8	B8_N0	Column I/O	8	B8_N0	Dedicated Clock	8	B8_N0

Device... Pin Finder... Show only highlighted pins Show migration differences Export... Help

Before adding migration device

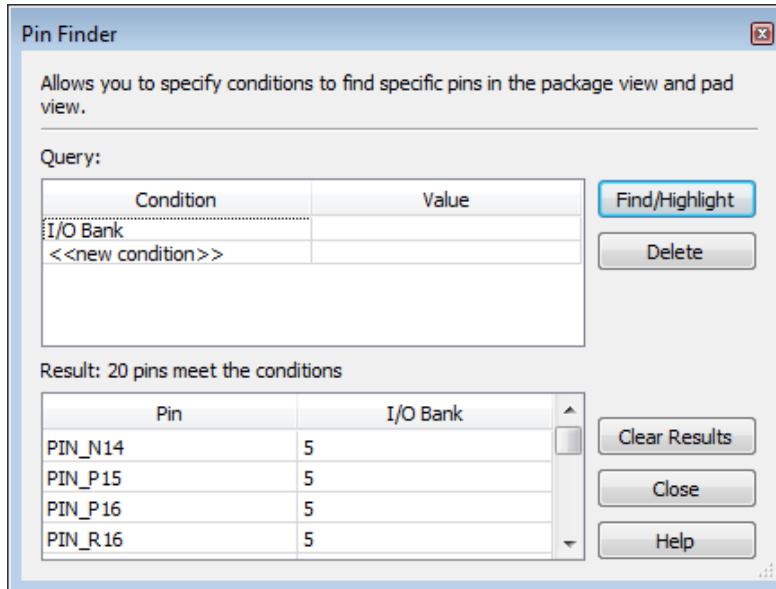


After adding migration device

More Pin Planner Features (1)

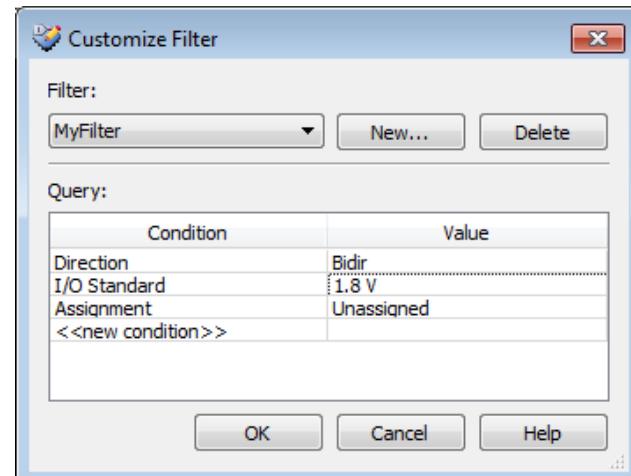
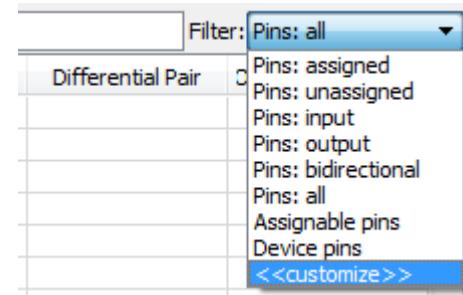
■ Pin Finder

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



■ Custom Filters

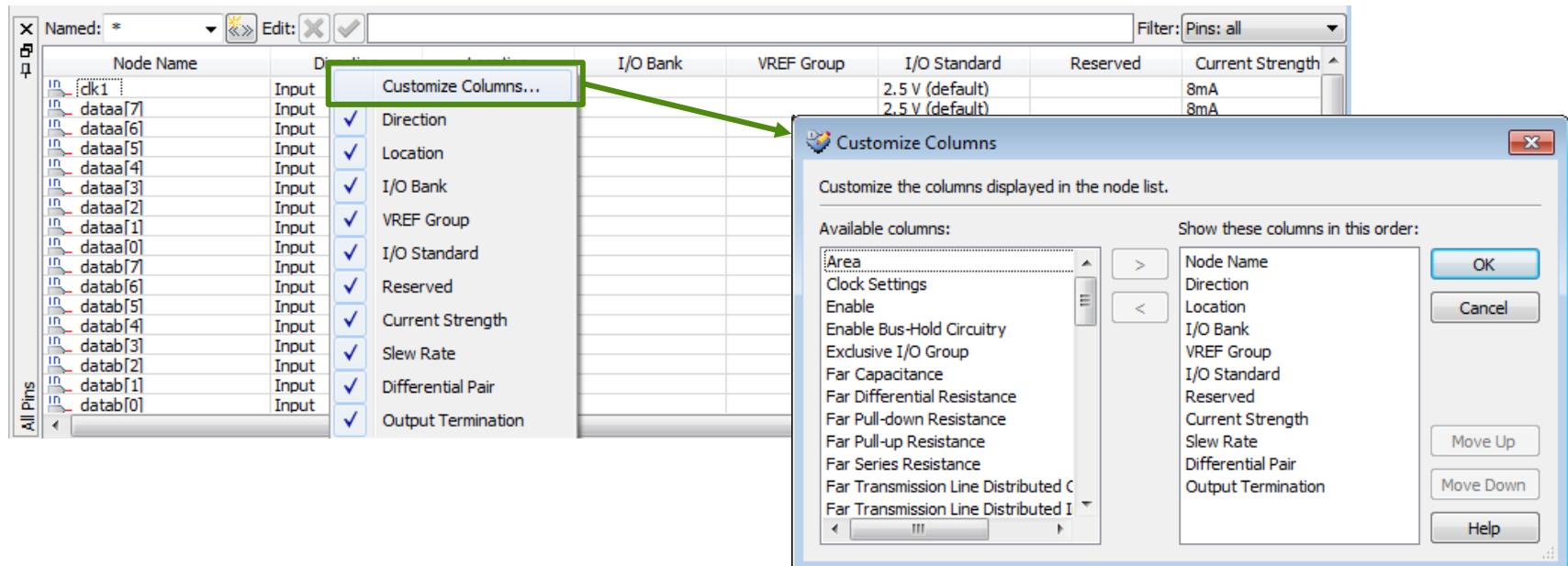
- Create custom filters for All Pins list



More Pin Planner Features (2)

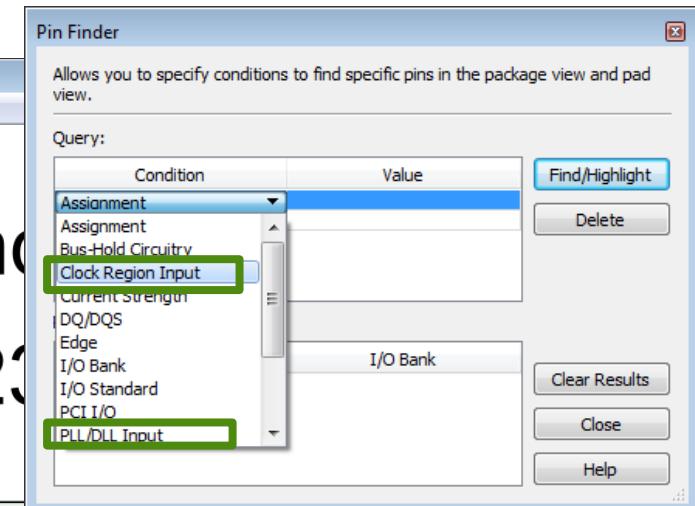
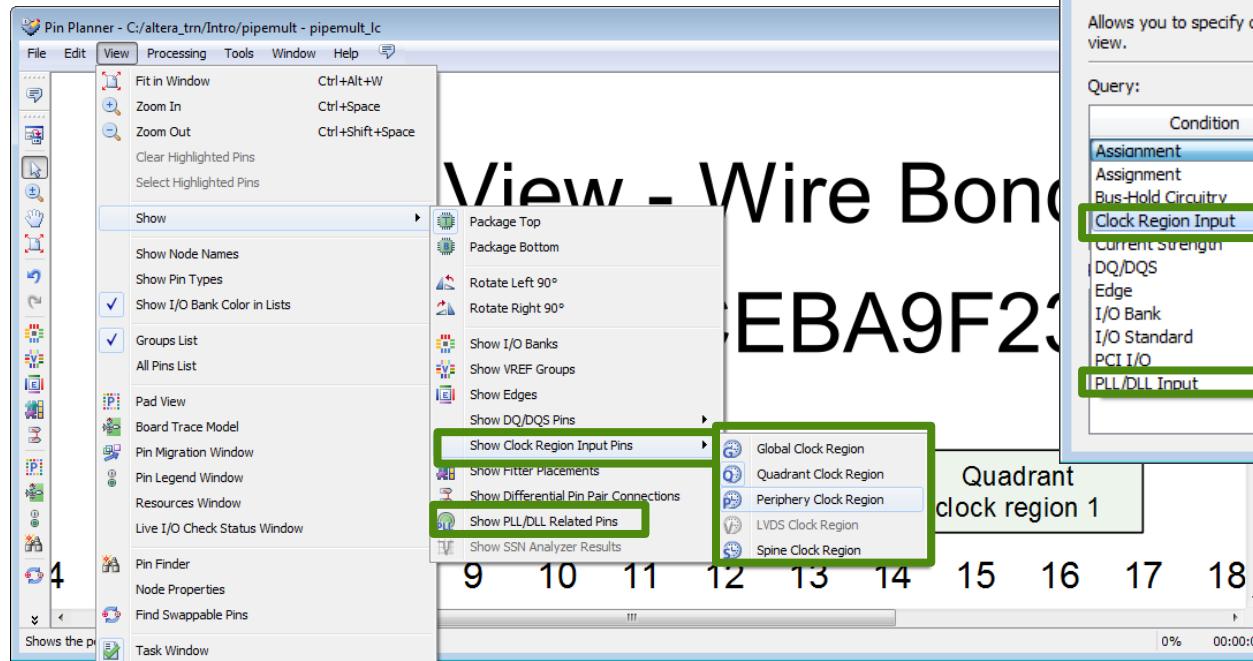
■ Customizable columns

- 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



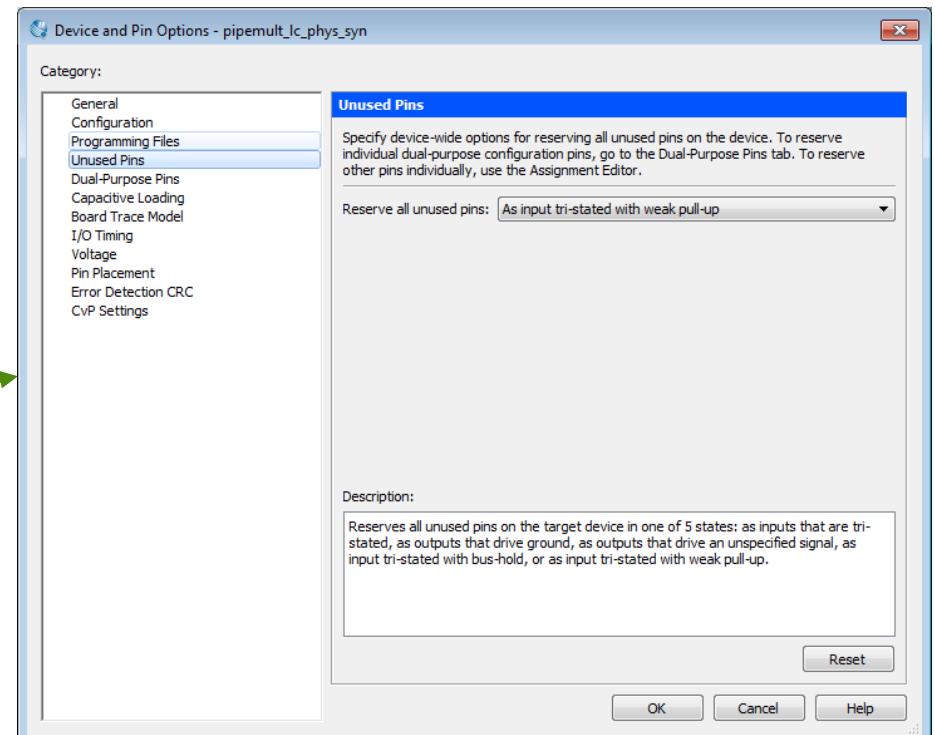
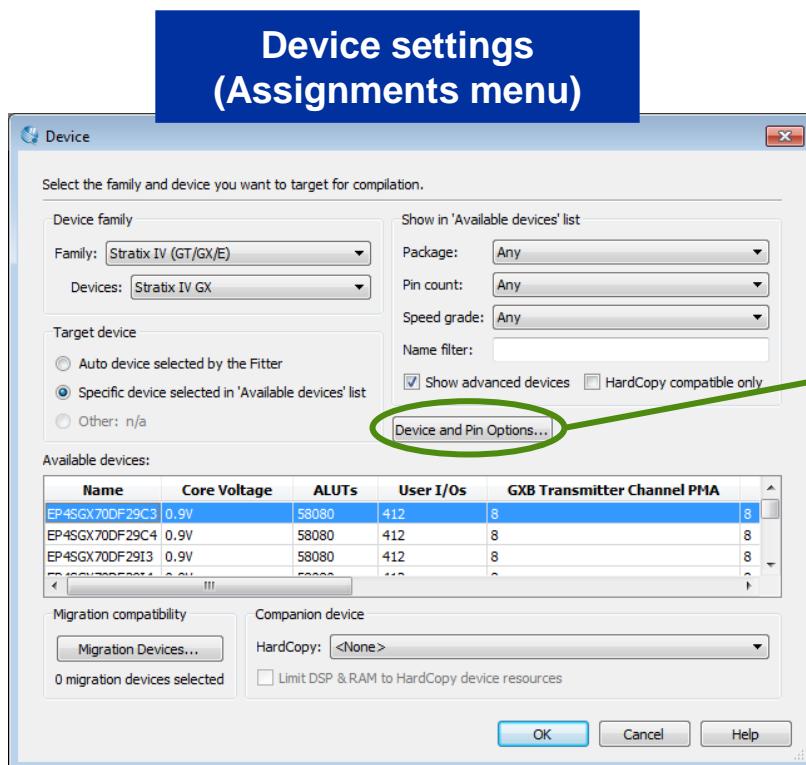
Pin Planner – Clock/PLL Support

- **Show Clock Region Input Pins view**
 - Shows the input pins in each region
- **Show PLL/DLL related pins**
- **Property windows shows clock region information**
- **Pin Finder allows searching of clock region input pins, PLL input/output pins**



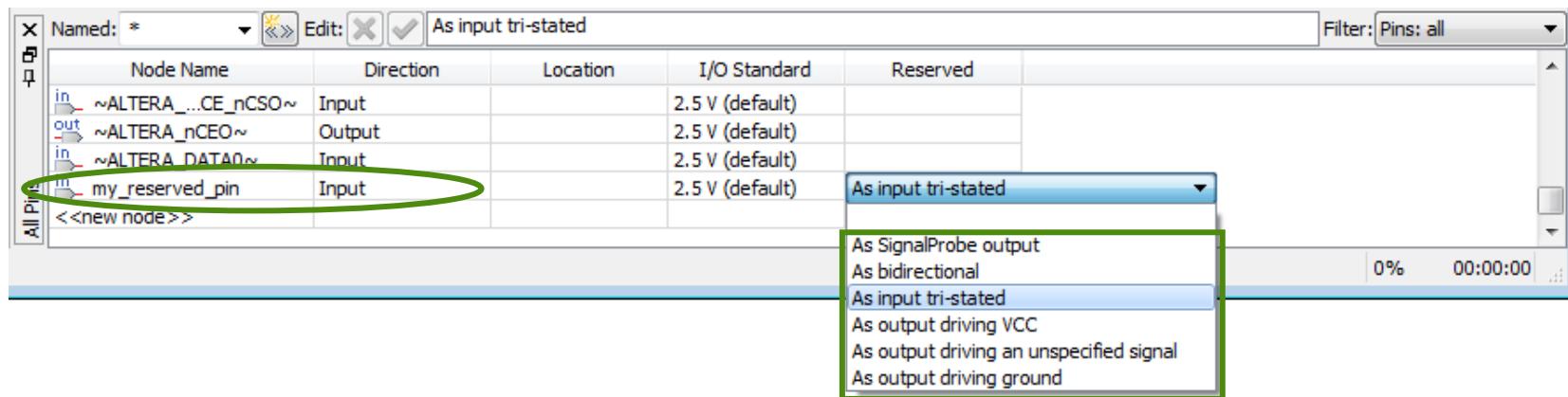
Global Pin Settings

- Select global settings for pins
- Assignments in Pin Planner or other tools have precedence over global settings



Reserved and Unused I/O Pins

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

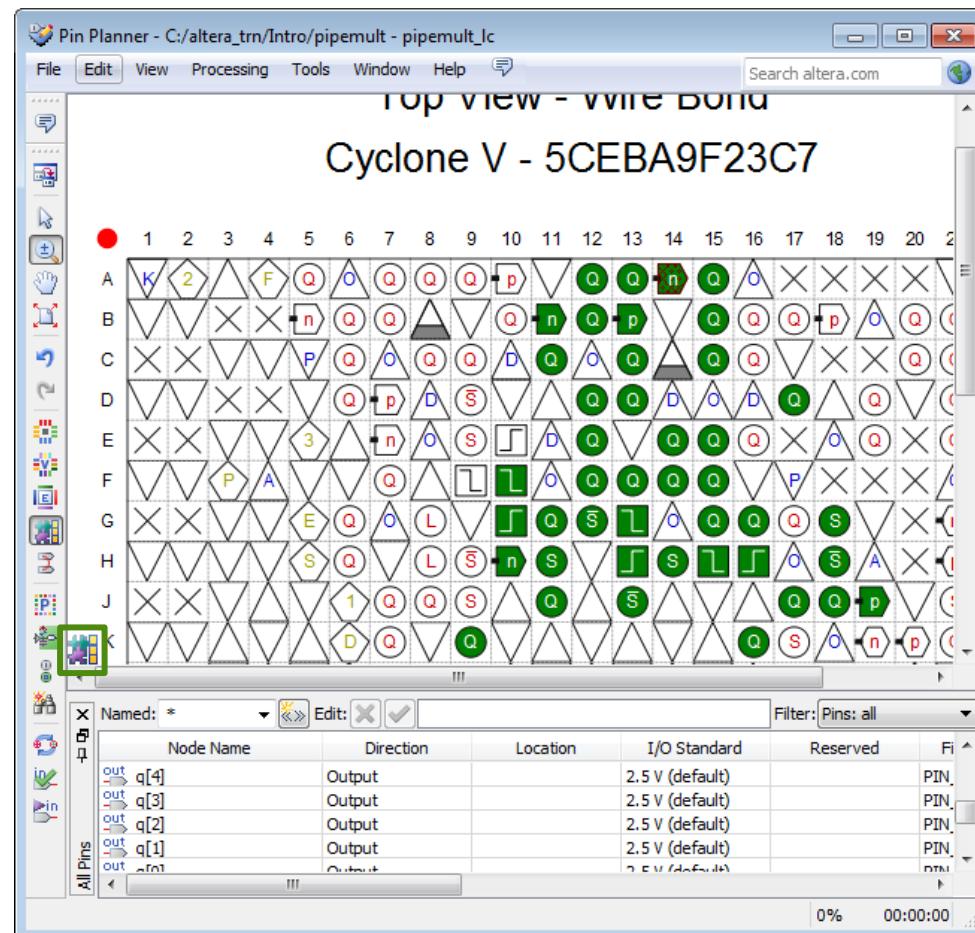


- Or right-click on pin in Package View and click Reserve Pins→ As...
- Pin name set to *user_reserve_<pin_number>*
- Set initial state of other unused pins in Device settings in Settings dialog box (see *previous slide*)

Show Fitter Placements

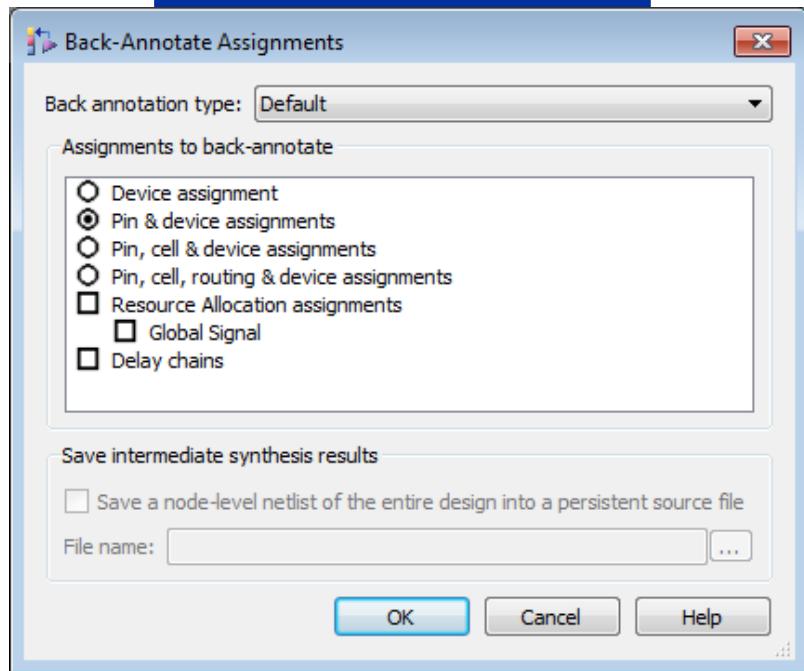
■ View I/O locations selected by the Fitter

View → Show
or in Toolbar



Back-Annotation

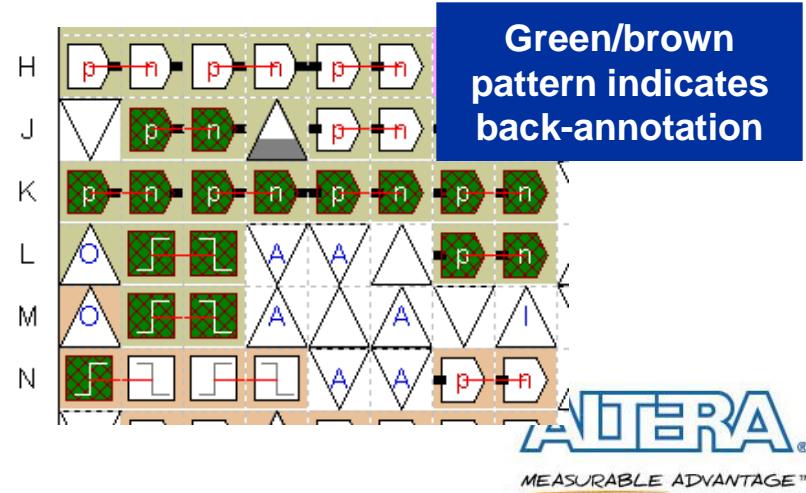
Assignments menu



■ Use to lock Fitter-chosen pin assignments for future compilations

- Copies device & resource locations chosen by fitter into QSF file
 - Pins
 - Logic
 - Routing

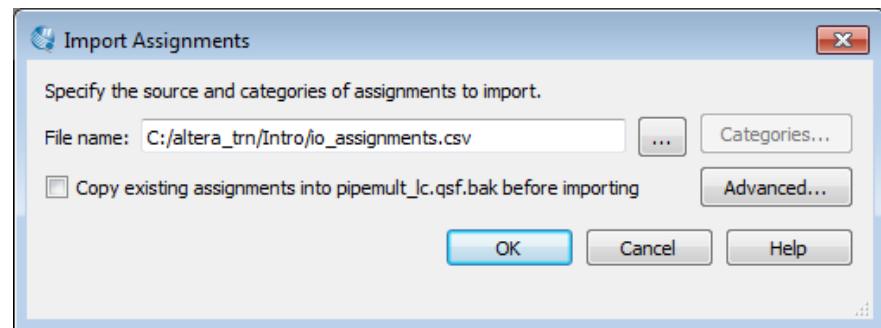
■ “Locks down” locations in Pin Planner



Other Methods: 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

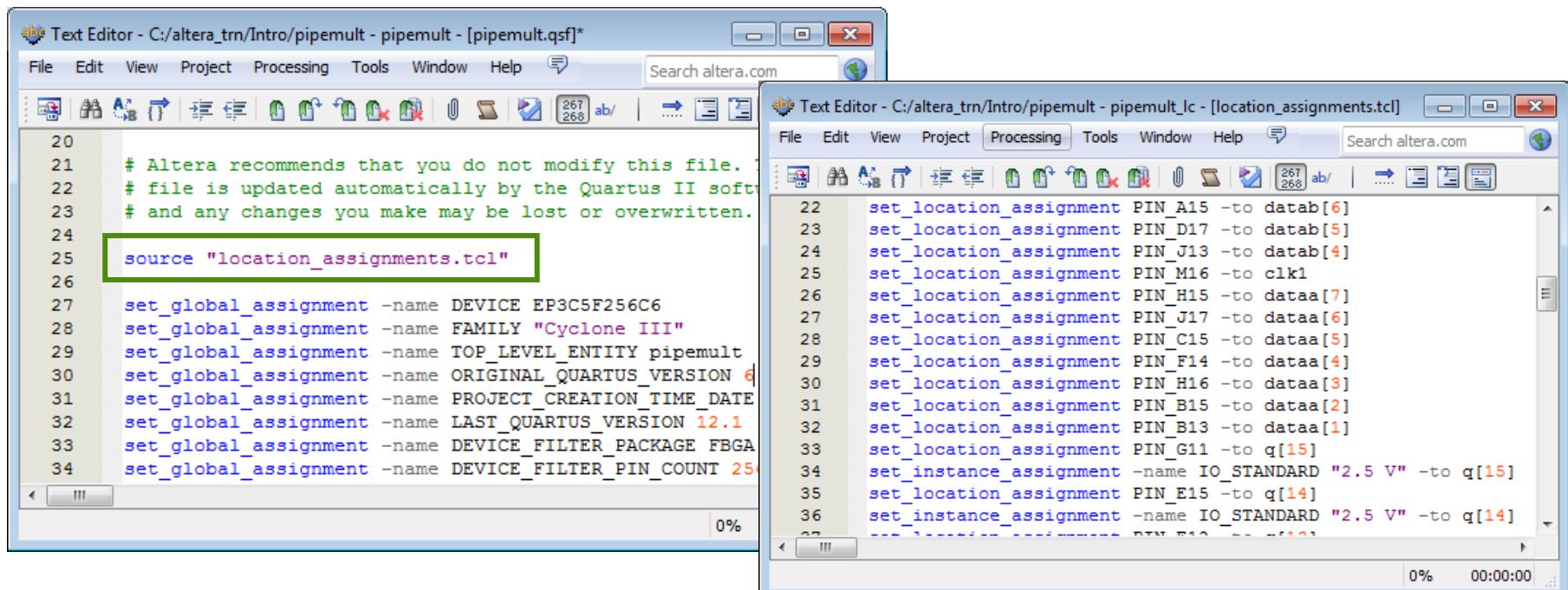
Pin Planner File menu or
Quartus II assignments menu



	A	B	C	D	E	F
1	To	Direction	Location	I/O Bank	VREF Group	I/O Standard
2	clk1	Input	PIN_E15	6	B6_N0	1.8 V
3	dataa[7]	Input	PIN_B14	7	B7_N0	2.5 V
4	dataa[6]	Input	PIN_A14	7	B7_N0	2.5 V
5	dataa[5]	Input	PIN_B13	7	B7_N0	2.5 V
6	dataa[4]	Input	PIN_A12	7	B7_N0	2.5 V

Other Methods: 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



The image shows two windows of the Altera Text Editor. The left window displays a QSF (Quartus Settings File) with the following content:

```
20
21  # Altera recommends that you do not modify this file.
22  # file is updated automatically by the Quartus II software
23  # and any changes you make may be lost or overwritten.
24
25  source "location_assignments.tcl"
26
27  set_global_assignment -name DEVICE EP3C5F256C6
28  set_global_assignment -name FAMILY "Cyclone III"
29  set_global_assignment -name TOP_LEVEL_ENTITY pipemult
30  set_global_assignment -name ORIGINAL_QUARTUS_VERSION 6
31  set_global_assignment -name PROJECT_CREATION_TIME_DATE
32  set_global_assignment -name LAST_QUARTUS_VERSION 12.1
33  set_global_assignment -name DEVICE_FILTER_PACKAGE FBGA
34  set_global_assignment -name DEVICE_FILTER_PIN_COUNT 25
```

The line 'source "location_assignments.tcl"' is highlighted with a green box. The right window displays a Tcl script with the following content:

```
22  set_location_assignment PIN_A15 -to datab[6]
23  set_location_assignment PIN_D17 -to datab[5]
24  set_location_assignment PIN_J13 -to datab[4]
25  set_location_assignment PIN_M16 -to clk1
26  set_location_assignment PIN_H15 -to dataaa[7]
27  set_location_assignment PIN_J17 -to dataaa[6]
28  set_location_assignment PIN_C15 -to dataaa[5]
29  set_location_assignment PIN_F14 -to dataaa[4]
30  set_location_assignment PIN_H16 -to dataaa[3]
31  set_location_assignment PIN_B15 -to dataaa[2]
32  set_location_assignment PIN_B13 -to dataaa[1]
33  set_location_assignment PIN_G11 -to q[15]
34  set_instance_assignment -name IO_STANDARD "2.5 V" -to q[15]
35  set_location_assignment PIN_E15 -to q[14]
36  set_instance_assignment -name IO_STANDARD "2.5 V" -to q[14]
37  ...
```

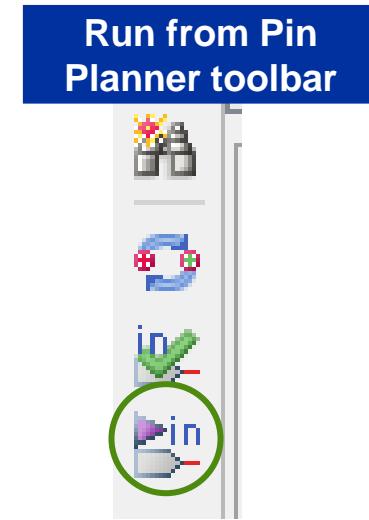
Verifying I/O Assignments

■ 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)
 - PCI clamping diode
 - Toggle rate



Processing menu →
Start → Start I/O Assignment Analysis
or Tasks window

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 V_{CCIO}

■ I/O connected to logic

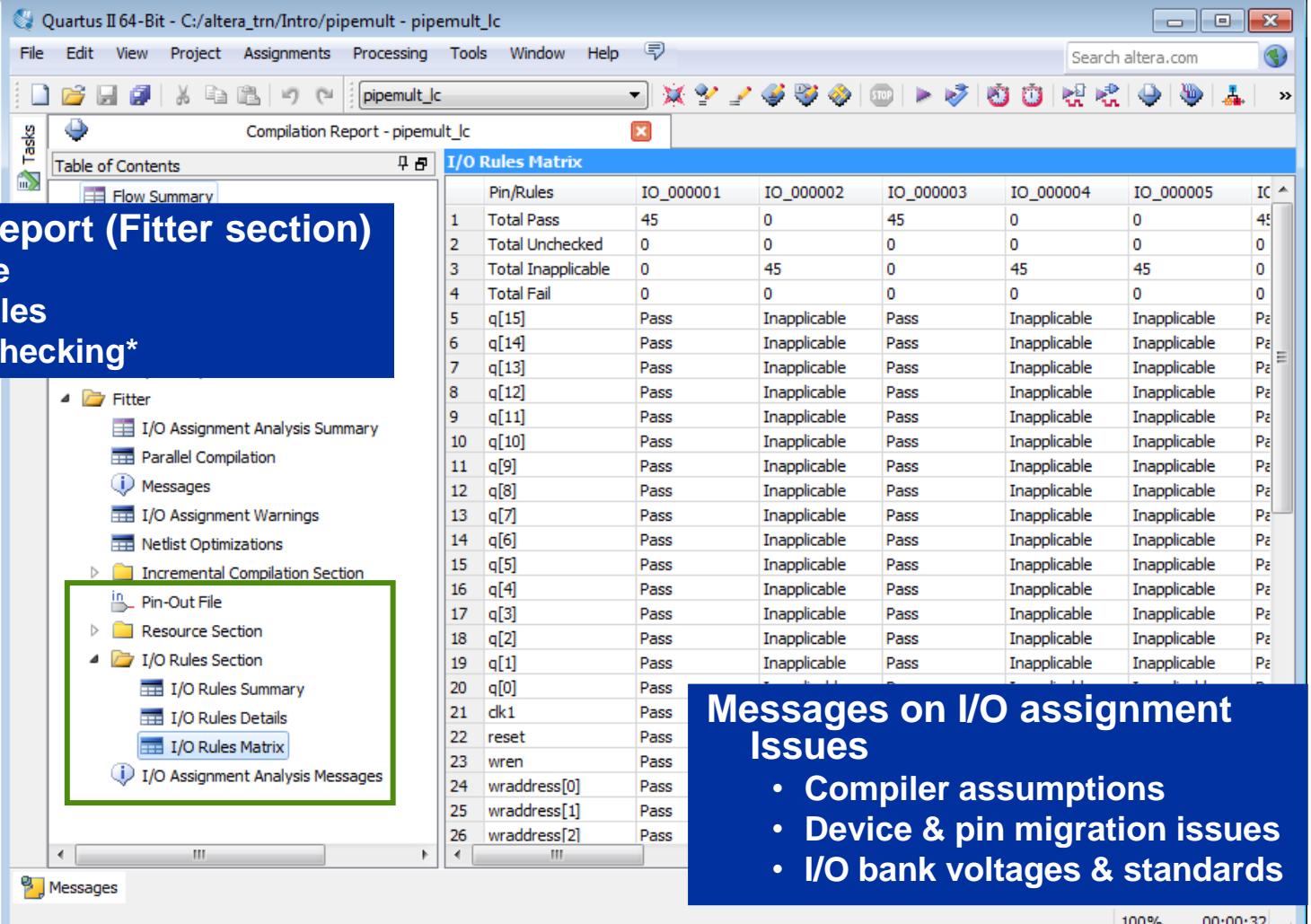
- Checks I/O locations & constraints with respect to other I/O, I/O banks, & internal resources
- e.g. PLL must be driven by a dedicated clock input pin

Note: When working with design files, synthesize design before running I/O Assignment Analysis

I/O Assignment Analysis Output

Compilation Report (Fitter section)

- Pin-out file
- I/O pin tables
- I/O rules checking*



Pin/Rules	IO_000001	IO_000002	IO_000003	IO_000004	IO_000005	IC
1 Total Pass	45	0	45	0	0	45
2 Total Unchecked	0	0	0	0	0	0
3 Total Inapplicable	0	45	0	45	45	0
4 Total Fail	0	0	0	0	0	0
5 q[15]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass
6 q[14]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass
7 q[13]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass
8 q[12]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass
9 q[11]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass
10 q[10]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass
11 q[9]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass
12 q[8]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass
13 q[7]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass
14 q[6]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass
15 q[5]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass
16 q[4]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass
17 q[3]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass
18 q[2]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass
19 q[1]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass
20 q[0]	Pass					
21 clk1	Pass					
22 reset	Pass					
23 wren	Pass					
24 wraddress[0]	Pass					
25 wraddress[1]	Pass					
26 wraddress[2]	Pass					

Messages on I/O assignment Issues

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

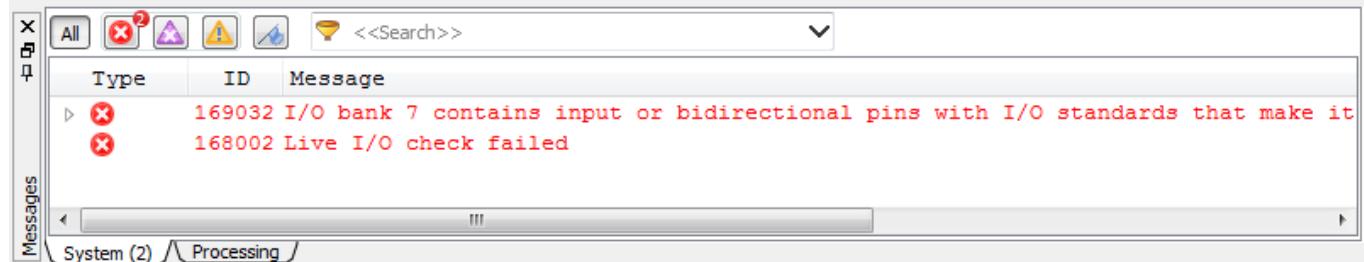
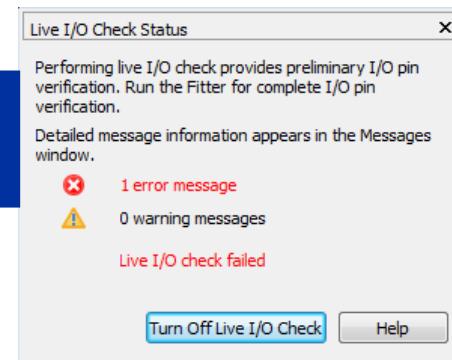
Live I/O Checking

- Perform limited I/O checks as assignments are made
- Status window alerts to failing assignments
 - Errors detailed in Messages window and Live I/O Check Status window
- Full I/O Assignment Analysis still required

Turn on or off
in Pin Planner
toolbar



In Pin Planner, View
menu → Live I/O
Check Status Window

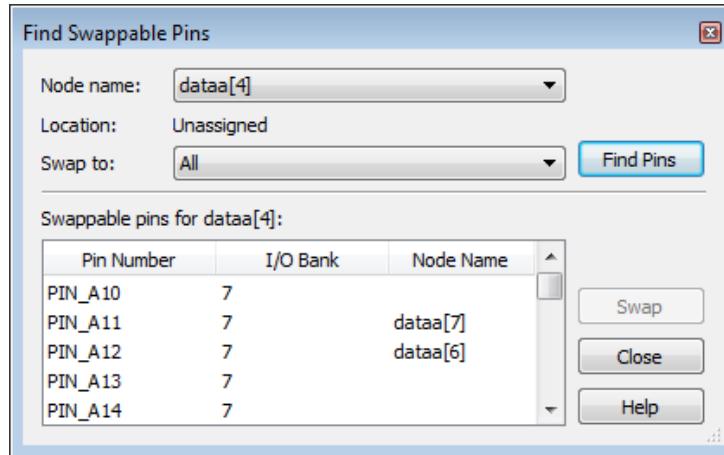


Swappable Pins



- Find compatible pins for swapping locations
- Works directly with Live I/O Check
 - Live I/O Check must be turned on
- Useful for last minute I/O location changes

View menu → Find Swappable Pins, Toolbar, or right-click menu



Selected pin
highlighted in
Package view

■ Completed design

- Run full compilation
- Enable option to run I/O Assignment Analysis before fitting

■ Incomplete design with completed top-level design file

- Run I/O Assignment Analysis on design

■ Incomplete or no design files

- Use early I/O planning methodology

Test Your Knowledge: I/O Planning

- What are some methods to assign pins using the Pin Planner?

A.

- What is the name of the real-time IO verification tool in Pin Planner?

A.

Exercise 5 Demonstration

*Demo should open automatically click
the link above if it doesn't*

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 of creating and managing pin assignments
- I/O Assignment Analysis helps validate a device pin-out without performing a full compilation
- Live I/O checking runs a subset of checks as assignments are made
- Pin validation can be completed during any point in design development

■ Quartus II Handbook chapters

- *I/O Management* (Volume 2)
- *Signal Integrity Analysis with Third-Party Tools* (Volume 2)
- *Mentor Graphics PCB Design Tools Support* (Volume 2)
- *Cadence PCB Design Tools Support* (Volume 2)

■ Training courses and demonstrations

- Online training: *I/O System Design*
- Online training: *Advanced I/O System Design*

Quartus II Software Design Series: Foundation

Programming & Configuration



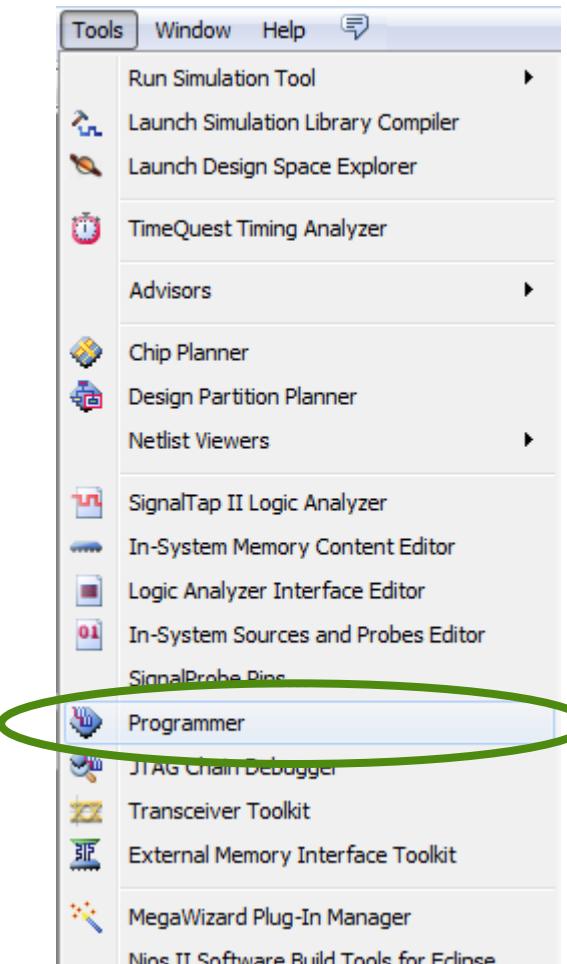
© 2013 Altera Corporation—Public



Programming Section Objectives

- Define the difference between the types of programming files generated by the Quartus II software
- Program a device
- Convert from one type of programming file to another

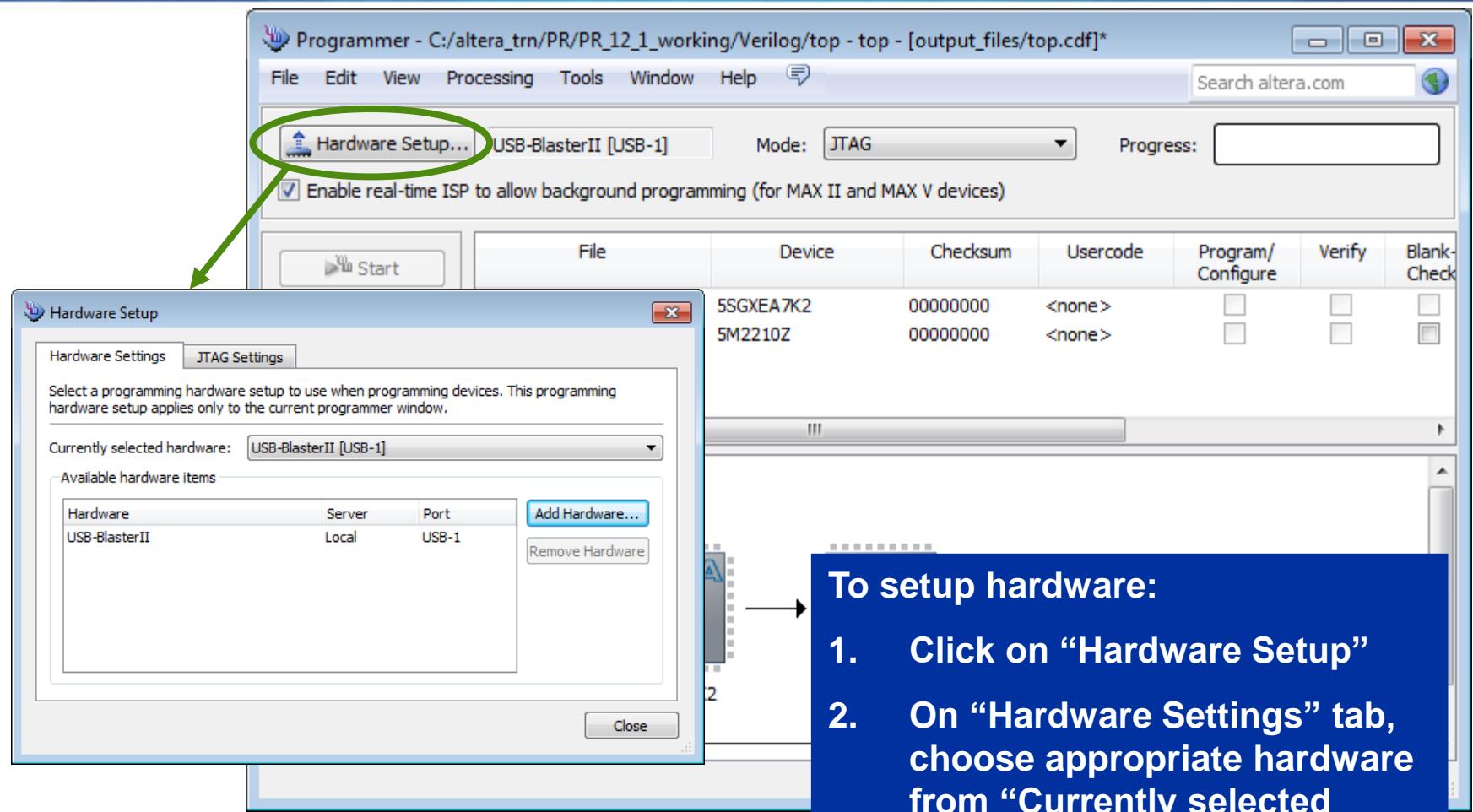
■ Use to configure Altera devices in-system



- Click on Programmer Icon on the toolbar
- or
- Tools menu→ Programmer



Setting Up the Programming Hardware



To setup hardware:

1. Click on “Hardware Setup”
2. On “Hardware Settings” tab, choose appropriate hardware from “Currently selected hardware” dropdown

Note: Add hardware as needed

Supported Programming Files

■ **.SOF (SRAM Object File)**

- Used to configure FPGAs directly from Quartus II software through download cable

■ **.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

■ **.JIC (JTAG Indirect Configuration File)**

- Used to program EPCS (Altera serial configuration) devices through their dedicated configuration interface with FPGAs

JTAG Chain

Programmer - C:/altera_trn/PR/PR_12_1_working/Verilog/top - top - [output_files/top.cdf]*

File Edit View Processing Tools Window Help Search altera.com

Hardware Setup... USB-BlasterII [USB-1] Mode: JTAG Progress:

Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Sec. Bi
output_files/top.sof	5SGXEA7K2F40	08A4A5C2	FFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<none>	5M2210Z	00000000	<none>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<none>	MY_DEVICE	00000000	<none>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

When adding files, the device for that file is automatically chosen

Enable programming

Change the order of the device chain

JTAG Chain:

- Can consist of combination of Altera FPGAs, CPLDs and configuration devices as well as non-Altera (or user) devices*
- Devices in the chain will be programmed from top down

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Sec. Bi
output_files/top.sof	5SGXEA7K2F40	08A4A5C2	FFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<none>	5M2210Z	00000000	<none>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<none>	MY_DEVICE	00000000	<none>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

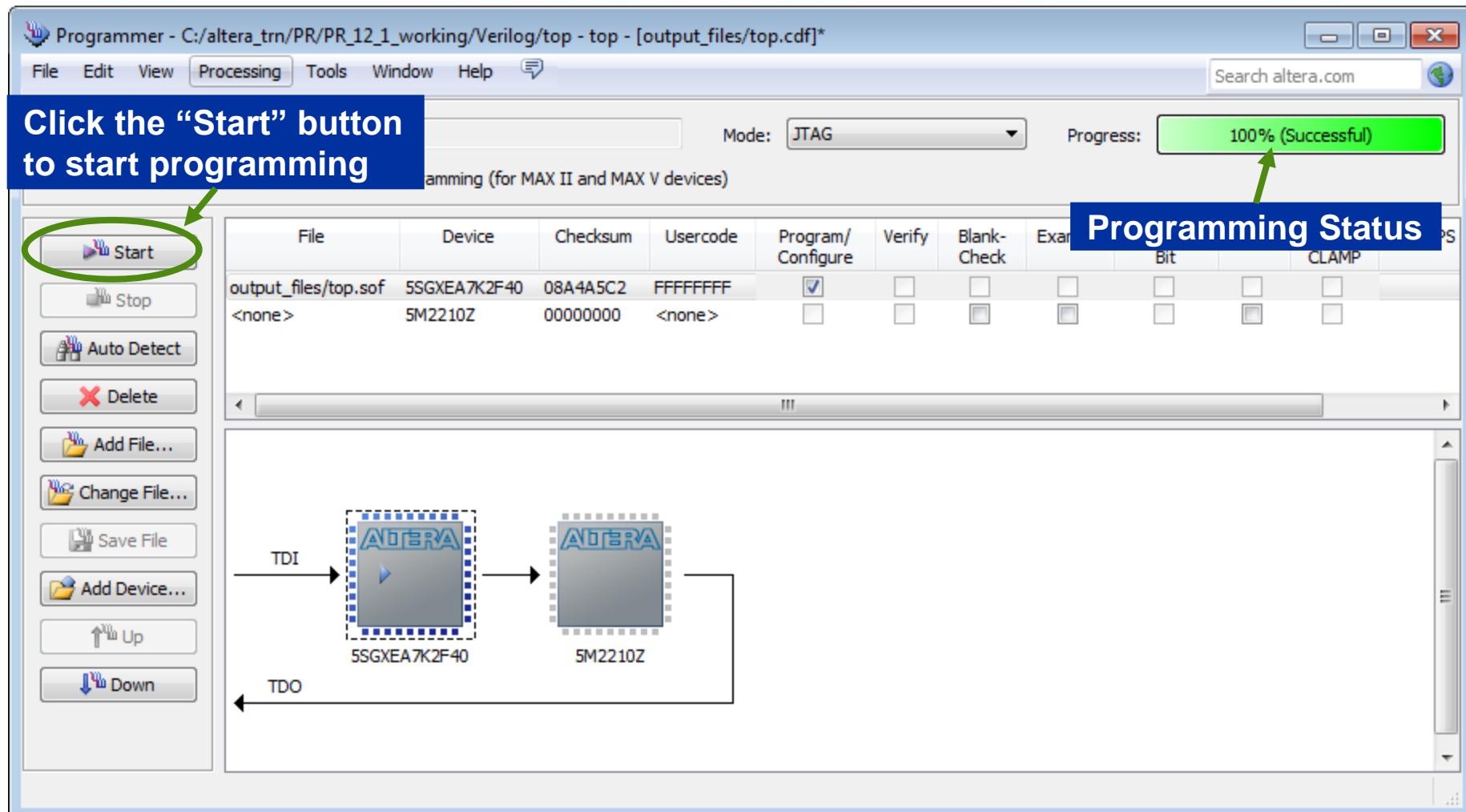
Programming Options

The screenshot shows the Altera Programmer interface. On the left, a blue sidebar contains the text: "These options only apply to CPLDs or configuration devices." Below this is a logic diagram showing a "Down" button connected to a "TDO" pin, with the device identifier "5SGXEA7K2F40" above it. The main window displays a table of programming options for three devices. The "Program/Configure" column for the first device has a checked checkbox. The "Verify", "Blank-Check", "Examine", "Security Bit", and "Erase" columns for all three devices are highlighted with a green box. A legend on the right maps these options to their descriptions.

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP	IPS File
output_files/top.sof	5SGXEA7K2F40	08A4A5C2	FFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
<none>	5M2210Z	00000000	<none>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<none>	MY_DEVICE	00000000	<none>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

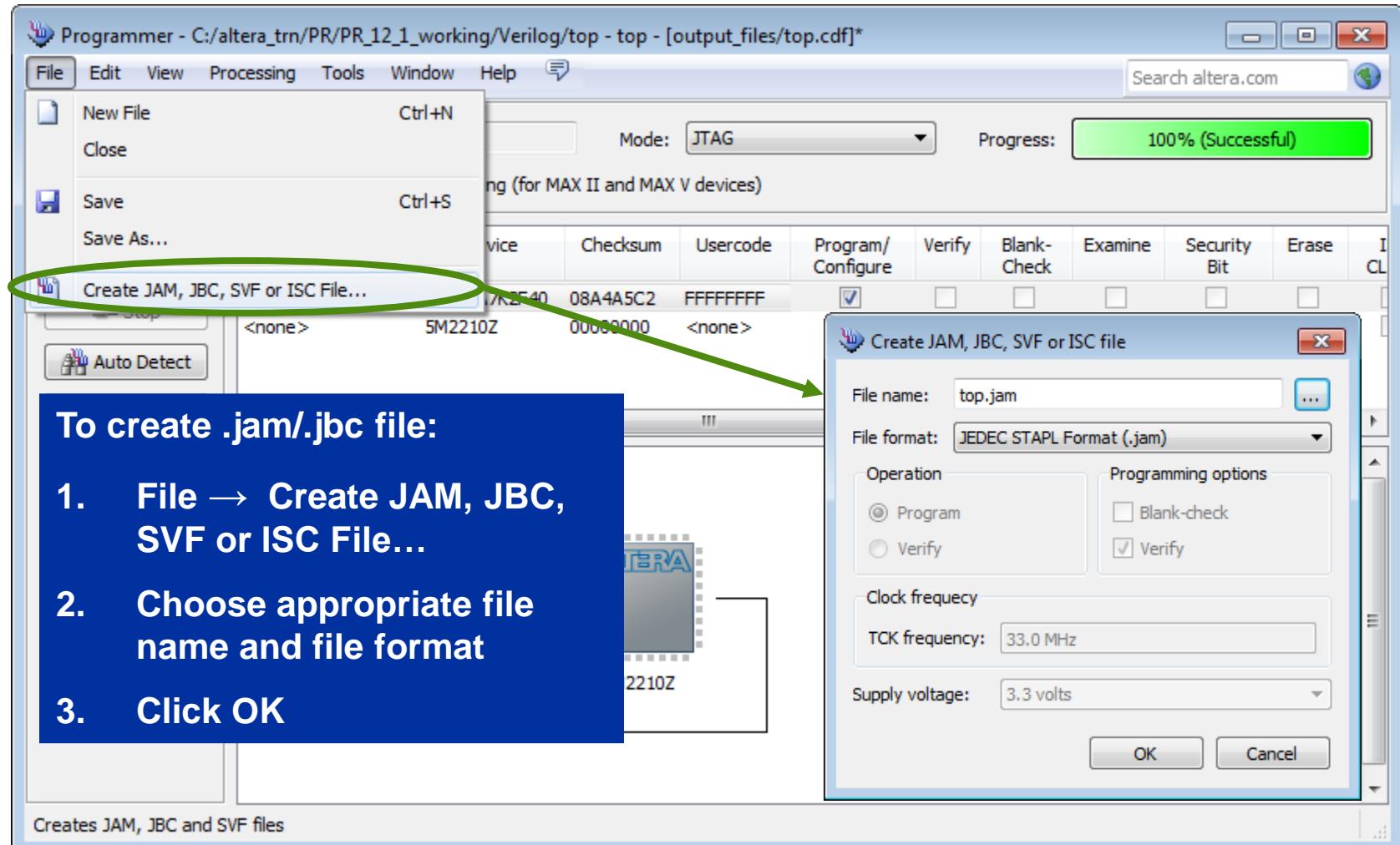
Option	Description
Verify	Verifies contents of against respective programming files
Blank-Check	Check whether the device is blank
Examine	Reads back contents. Can save examine data as .pof file.
Security Bit	Protect CPLD from being examined.
Erase	Erase the contents of devices

Programming



Creating .jam/.jbc file

JAM/JBC files allow external processors or test equipment to program devices



To create .jam/.jbc file:

1. File → Create JAM, JBC, SVF or ISC File...
2. Choose appropriate file name and file format
3. Click OK

Programming .jic File

Programmer - C:/altera_trn/Intro/pipemult - pipemult - [Chain1.cdf]*

File Edit View Processing Tools Window Help

Hardware Setup... No Hardware Mode: JTAG Progress:

Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Se
Factory default enhanced SFL image	EP4CE6	0000000000000000	0000000000000000	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
output_file.jic	EPCS16	1A5B4633	0000000000000000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Start
 Stop
 Auto Detect
 Delete
 Add File...
 Change File...
 Save File
 Add Device...
 Up
 Down

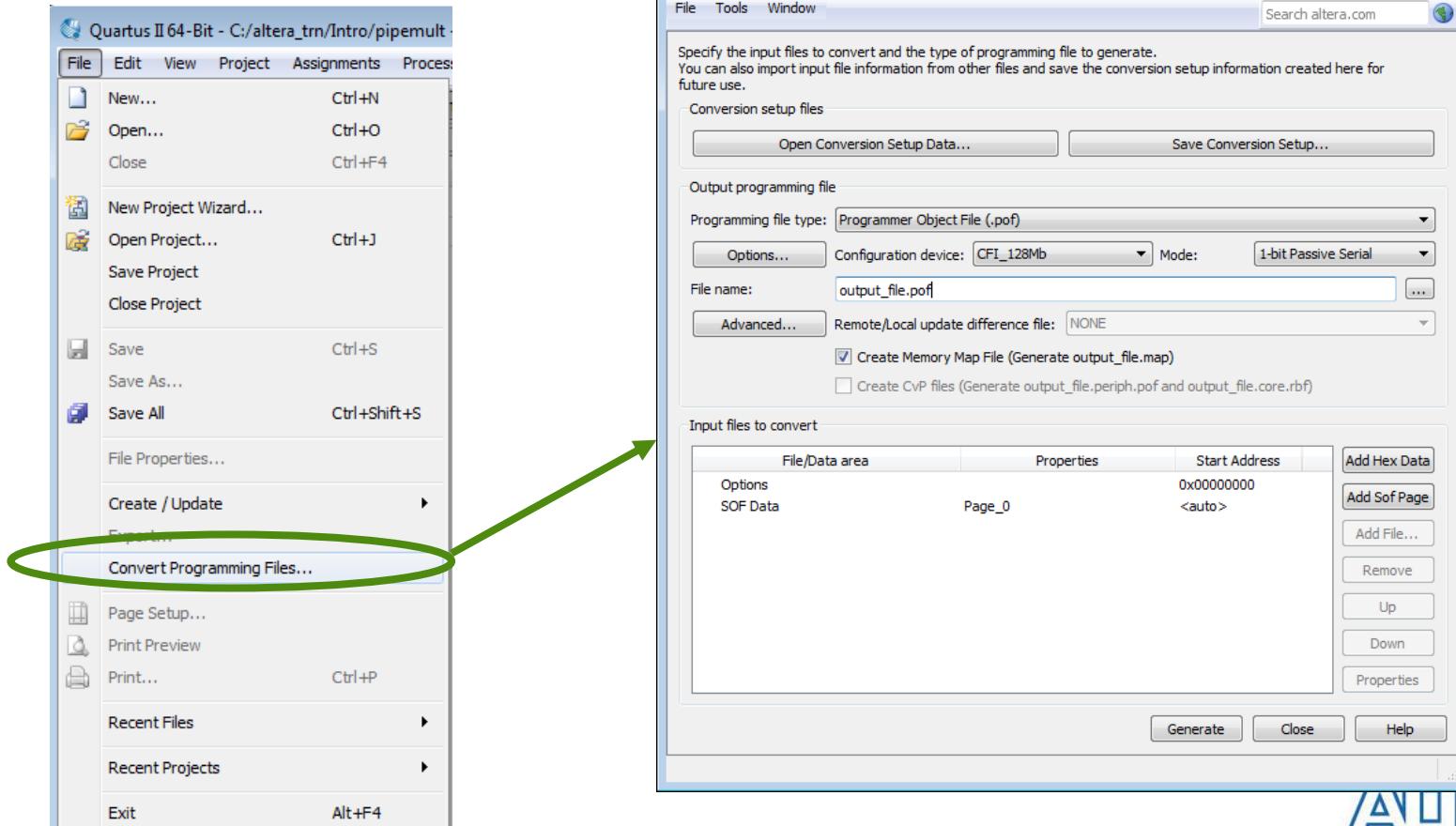
JIC files used to program EPCS configuration devices (which do not have JTAG pins)

Programming File Conversion

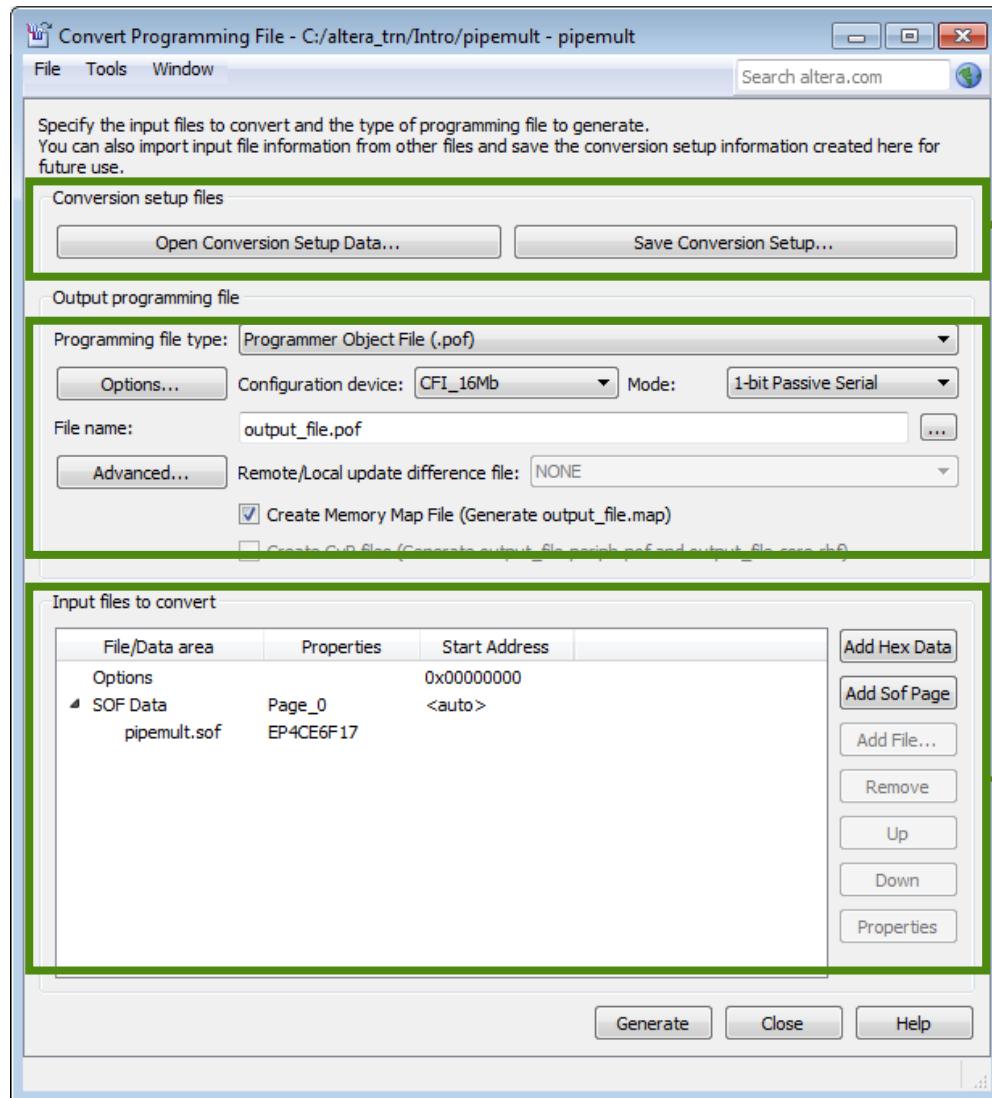
- **Quartus II Assembler automatically generates .SOF files for any FPGA**
 - May generate additional single-device programming file types from Device Settings dialog box (Assignments menu → Device → Device and Pin Options → Programming Files)
- **.SOF files can only be used by the Quartus II Programmer to directly configure FPGA**
- **Other configuration solutions require converting the .SOF file(s) into other file formats**
 - .JIC files
 - Multi-FPGA programming files

Converting Programming Files

■ Convert .sof and .pof files to other supported programming files



Convert Programming Files GU



Load or save file conversion setup

Select an optional programming file format

Select a configuration device and its supported mode

Select more options for the configuration device

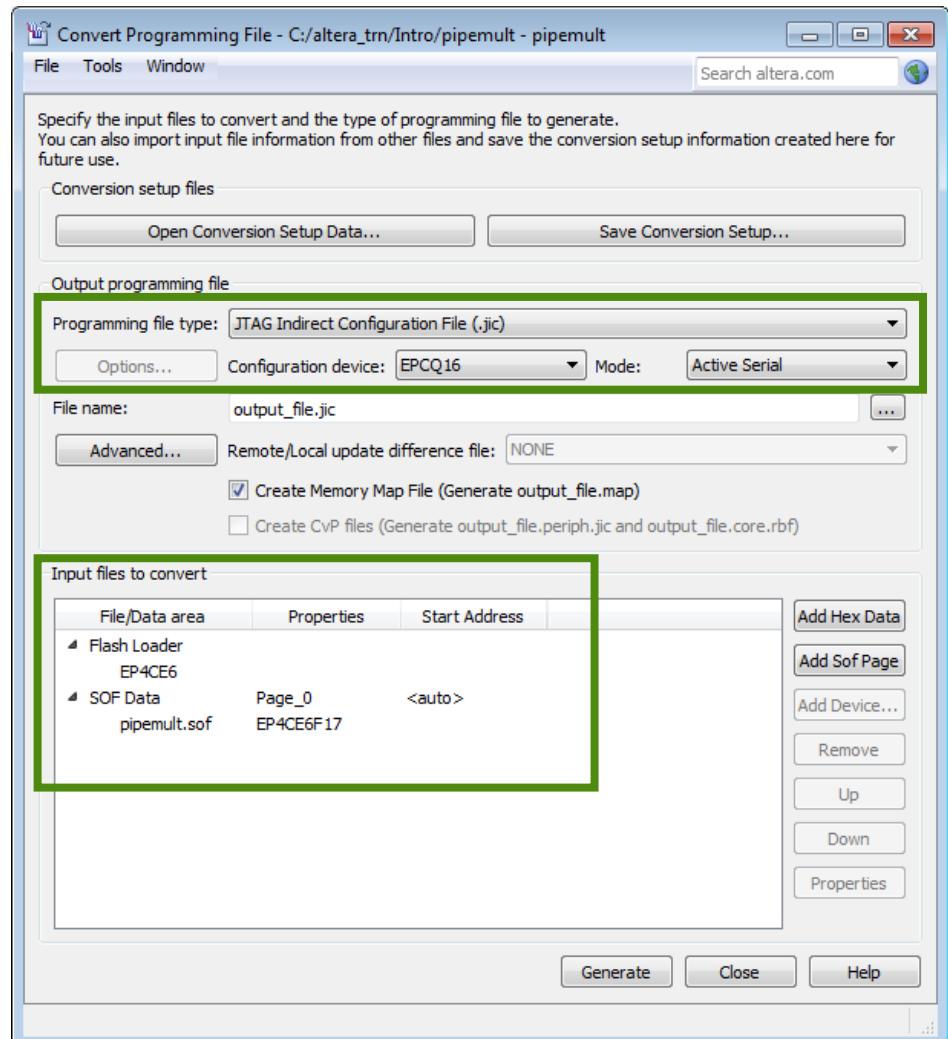
Add input configuring/ programming file(s)

EPCS Programming - .jic generation



To create the .jic file:

- 1. Select .jic as output file type**
- 2. Highlight Flash Loader**
 - Click “Add Device”**
 - Browse for FPGA**
- 3. Click “Generate”**



Test Your Knowledge: Programming

- 1. What are some of the types of programming files generated by the Quartus II software?**

Summary

- Use the Quartus II Programmer to program and configure Altera devices for in-system testing
- Use the file conversion utility to generate additional programming file types

Programming/Configuration Support Resources

■ Quartus II Handbook Chapters

- *Quartus II Programmer* (Volume 3)

■ Training & Demonstrations

- Online training: [Configuring Altera FPGAs](#)
- Online training: [Debugging JTAG Chain Integrity](#)

Class Summary

- **Quartus II Projects**
- **Design Entry**
- **Quartus II Compilation**
- **Settings & Assignments**
- **I/O Planning**
- **Programming/Configuration**

Additional Quartus II Courses

■ Quartus II Software Debug and Analysis Tools

- Power Analysis
- SSN Analysis
- Debugging solutions
 - SignalProbe incremental routing
 - SignalTap II Embedded Logic Analyzer
 - In-System Sources & Probes
- In-System Memory Content Editor
- Chip Planner & Resource Property Editor
- System Console and Related Toolkits

■ Quartus II Software Design Series: Timing Analysis

- Create timing constraints to meet and optimize timing using TimeQuest TA
- Perform detailed timing analysis on an Altera device with TimeQuest TA

■ Advanced Timing Analysis with TimeQuest

- Automate constraining and analysis of FPGA designs using Tcl
- Constrain advanced types of interfaces and blocks

■ Timing Closure with the Quartus II Software

- Employ best practices to close timing using the tools available

■ Introduction to the Qsys System Integration Tool

- Integrate IP and custom logic into a Qsys system

Learn More Through Technical Training

Instructor-Led Training



With Altera's instructor-led training courses, you can:

- Learn from an experienced Altera technical training engineer (instructor)
- Complete hands-on exercises with guidance from an Altera instructor
- Ask questions and receive real-time answers from an Altera instructor
- Each instructor-led class is one or two days in length (8 working hours per day)

Virtual Classroom Training

With Altera's virtual classroom training:

- Get the best of both worlds!
- All the benefits of a live, instructor-led training class from the comfort of your home or office

Online Training



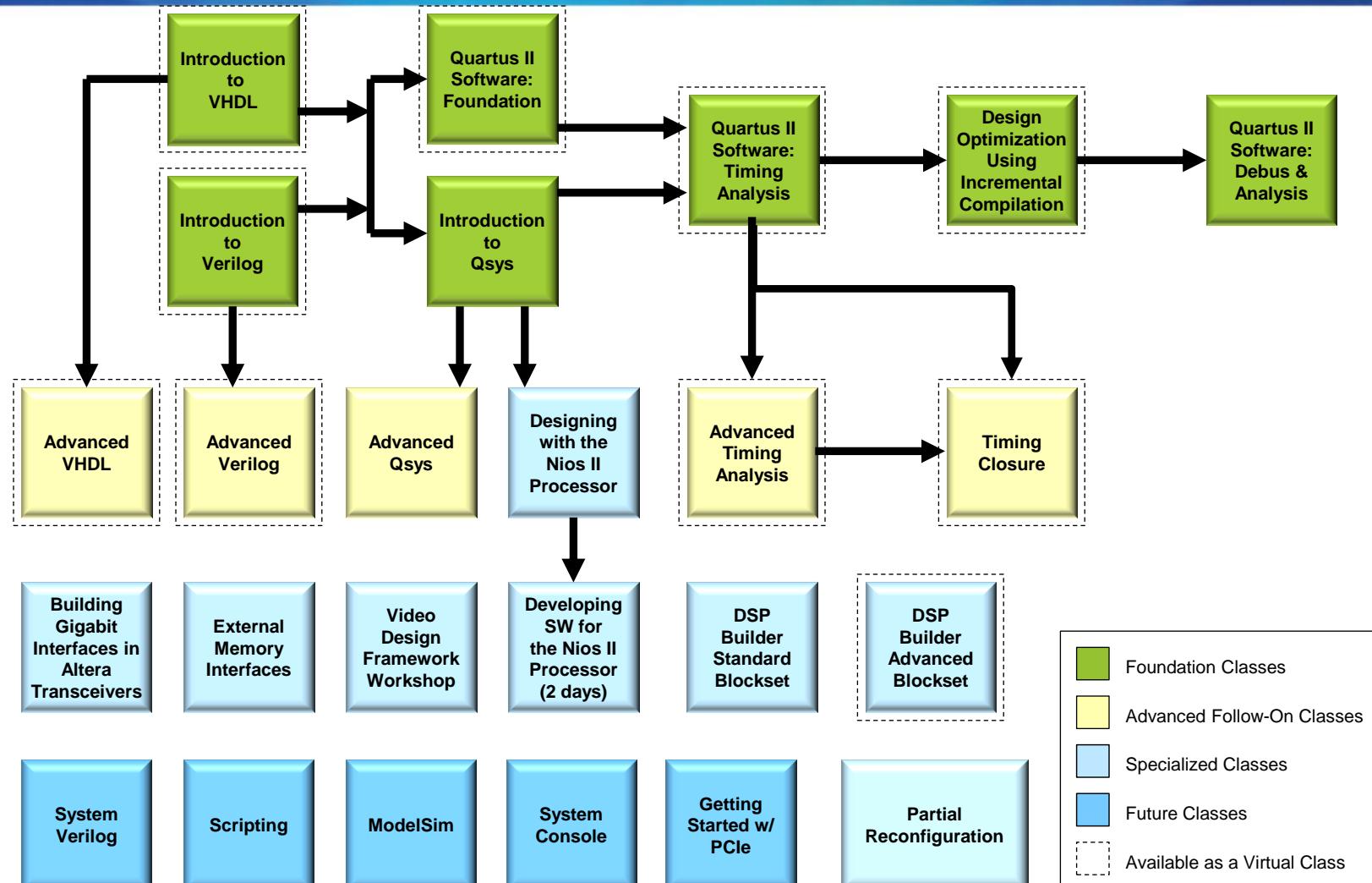
With Altera's online training courses, you can:

- Take a course at any time that is convenient for you
- Take a course from the comfort of your home or office (no need to travel as with instructor-led courses)
- Each online course takes approximate one to three hours to complete

<http://www.altera.com/training>

View training class schedule and register for a class

Instructor-Led and Virtual Training Curriculum



- **Reference Quartus II software on-line help**
- **Quartus II Handbook**
- **World-wide web: <http://www.altera.com>**
 - Search for answers to problems with Knowledge Database
 - Download literature
 - View design examples
 - View online trainings
- **MySupport: <http://www.altera.com/mysupport>**
- **Field applications engineers: contact your local Altera sales office**
- **Altera Wiki: www.alterawiki.com**
- **Altera Forum: www.alteraforum.com**
- **Intellectual Property Support**
 - <http://www.altera.com/support/ip/ips-index.html>

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!

Thank You



© 2013 Altera Corporation—Public

ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/legal.

ALTERA
MEASURABLE ADVANTAGE™