Peter the Great St. Petersburg Polytechnic University

Institute of Computer Science and Cybersecurity

Graduate School of Computer Technologies and Information Systems

Lecture: Creating a System Design with Platform Designer

Subject: Automation of discrete device design (in English)

Completed by student of group 5130901/10101 __________ Nepomnyaschiy M.T.
(signature)

Lecturer _____________________________ Antonov A.P.
(signature)

Saint Petersburg

2024

Table of contents

List of illustrations

Creating a System Design with Platform Designer

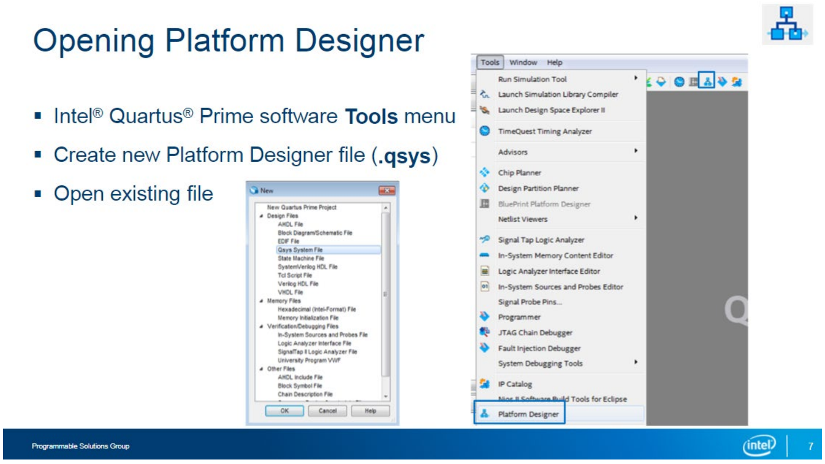
The lecture provides some information about Platform Designer usage and explains the examples of its usage on practice.

Introduction

The Platform Designer in Intel Quartus Prime comes in two versions, depending on whether you're using the Lite or Standard edition, or the Pro edition. This distinction significantly impacts the design process. In the Standard edition, a .qsys file captures all system information, requiring regeneration of RTL code for the entire system upon any parameter or connection changes, making it time-consuming for large designs. On the other hand, the Pro edition separates component information into .ip files, enabling selective regeneration and allowing incomplete components during system generation. This flexibility is particularly advantageous for ongoing development or team-based projects. The Pro edition not only maintains ease of use but also offers enhanced control and improved interactivity in system design, making it a substantial productivity boost compared to the Standard edition.

Part 6 – Platform Designer User Interface

6.1. Opening Platform Designer



Opening Platform Designer

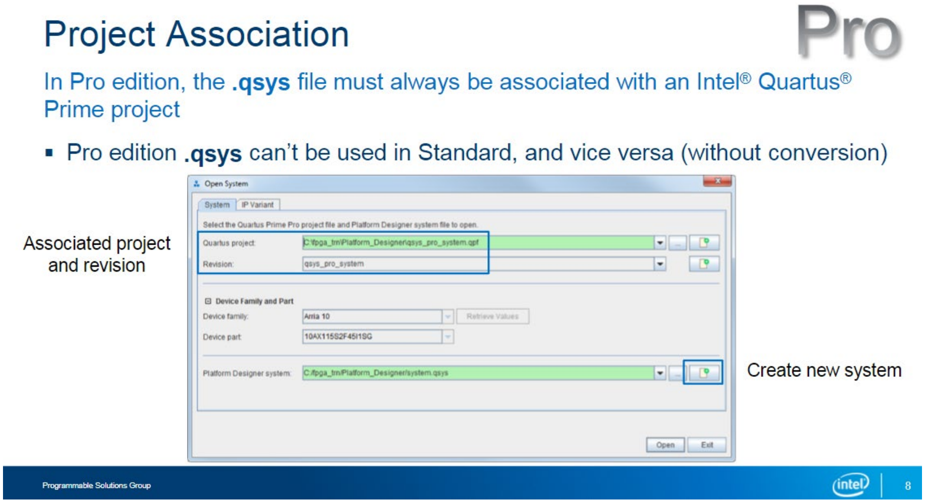
- Intel® Quartus® Prime software **Tools** menu
- Create new Platform Designer file (**.qsys**)
- Open existing file

New
New Quartus Prime Project
Design Files
AHDL File
Block Diagram/Schematic File
EDIF File
Qsys System File
State Machine File
SystemVerilog HDL File
Tcl Script File
Verilog HDL File
VHDL File
Memory Files
Hexadecimal (Intel-Format) File
Memory Initialization File
Verification/Debugging Files
In-System Sources and Probes File
Logic Analyzer Interface File
SignalTap II Logic Analyzer File
University Program VWF
Other Files
AHDL Include File
Block Symbol File
Chain Description File
OK
Cancel
Help
Tools
Window
Help
Run Simulation Tool
Launch Simulation Library Compiler
Launch Design Space Explorer II
TimeQuest Timing Analyzer
Advisors
Chip Planner
Design Partition Planner
BluePrint Platform Designer
Netlist Viewers
Signal Tap Logic Analyzer
In-System Memory Content Editor
Logic Analyzer Interface Editor
In-System Sources and Probes Editor
Signal Probe Pins...
Programmer
JTAG Chain Debugger
Fault Injection Debugger
System Debugging Tools
IP Catalog
Nios II Software Build Tools for Eclipse
Platform Designer
Programmable Solutions Group
intel
7

Figure 1 – Launching Platform Designer

To start creating a system design in Platform Designer, launch it from the Intel Quartus Prime software Tools menu. There's also a toolbar button you can use as a shortcut. You can also launch Platform Designer by selecting New from the File menu. From the New file dialog box, select Qsys System File as the design file type to create. If you have an existing Platform Designer system you want to modify, from the File menu, select Open and choose the .qsys file for your system.

6.2. Project Association



Project Association

In Pro edition, the **.qsys** file must always be associated with an Intel® Quartus® Prime project

- Pro edition **.qsys** can't be used in Standard, and vice versa (without conversion)

Open System

Select the Quartus Prime Pro project file and Platform Designer system file to open.

Quartus project: C:\fpga_trn\Platform_Designer\qsys_pro_system.qpf

Revision: qsys_pro_system

Device Family and Part

Device family: Arria 10

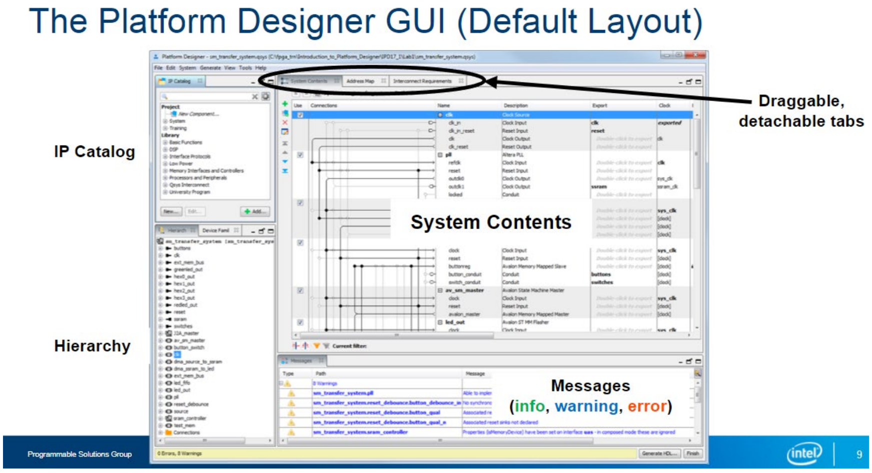
Device part: 10AX115S2F45I1SG

Platform Designer system: C:/fpga_trn/Platform_Designer/system.qsys

Figure 2 – Project Association

Using the Platform Designer tool is similar in the Standard and Pro editions, but in the Pro edition, your system design needs to be associated with an Intel Quartus Prime project. Unlike the Standard edition, which allows adding a design to any project easily, in the Pro edition, you start Platform Designer within a project or create a new one. Note that a .qsys file made in the Pro edition can't be used in the Standard edition, but the reverse is possible. If you switch from Pro to Standard, make a backup of the .qsys file before opening it in the Standard edition.

6.3. The Platform Designer GUI (Default view)

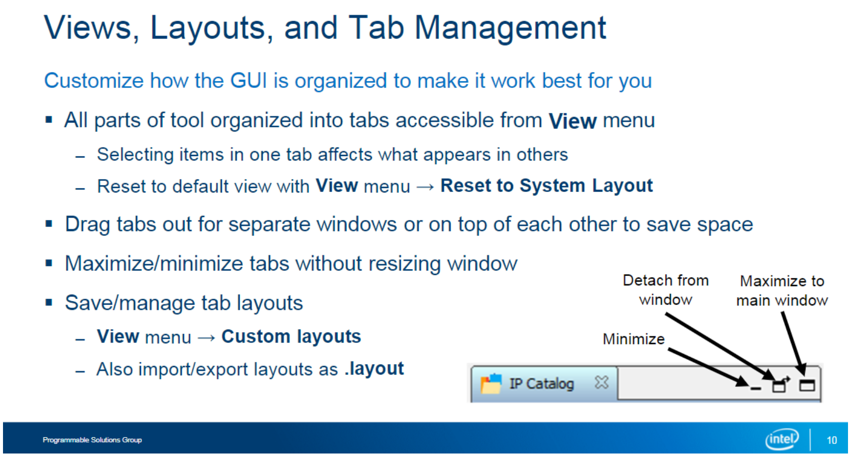


The Platform Designer GUI (Default Layout)

Figure 3 – GUI (Default Layout)

Here's an example of the Platform Designer UI displaying the main view and visible tabs. We'll delve deeper into these aspects later. The UI consists of tabs for constructing your system and analyzing results. You can enable or disable tabs as per your information requirements. While some windows overlap, viewing information in various ways and formats often enhances your comprehension of the system and its connections.

6.4. Views, Layouts, and Tab Management



Views, Layouts, and Tab Management

Customize how the GUI is organized to make it work best for you

- All parts of tool organized into tabs accessible from **View** menu
  - Selecting items in one tab affects what appears in others
  - Reset to default view with **View** menu → **Reset to System Layout**
- Drag tabs out for separate windows or on top of each other to save space
- Maximize/minimize tabs without resizing window
- Save/manage tab layouts
  - **View** menu → **Custom layouts**
  - Also import/export layouts as **.layout**

Figure 4 – Views, Layouts, and Tab Management

1. Enabling and Disabling Views and Tabs:
   - Access the View menu to enable or disable various views and tabs.
   - Click once to toggle, and a second time to restore the previous state.

*All tabs are interconnected, so actions in one tab affect others.

2. Reset and Reposition Tabs:
   - Restore all tabs to default view with "Reset to System Layout" from the View menu.
   - Click and drag tabs to reposition, allowing customization of the interface.
3. Tab Interactions and Customization:
   - Tabs can be linked or detached for efficient use across multiple monitors.
   - Drag tabs to stack or position them in the main interface for optimal display.
   - Minimize or maximize tabs using icons in the upper right corner.
4. Preserving and Exporting Layouts:
   - Platform Designer maintains the current tab layout even after system closure.
   - Save the current layout as a custom layout in the View menu.
   - Export the layout as a .layout file for reuse in different system designs.
   - Create multiple .layout files for diverse setups to enhance workflow optimization.

6.5. Basic Platform Designer System Building Flow

Basic Platform Designer System Building Flow

1. Select and configure components from IP Catalog
2. Connect component interfaces together
3. Review connections
4. Generate HDL for system

Figure 5 – System building flow

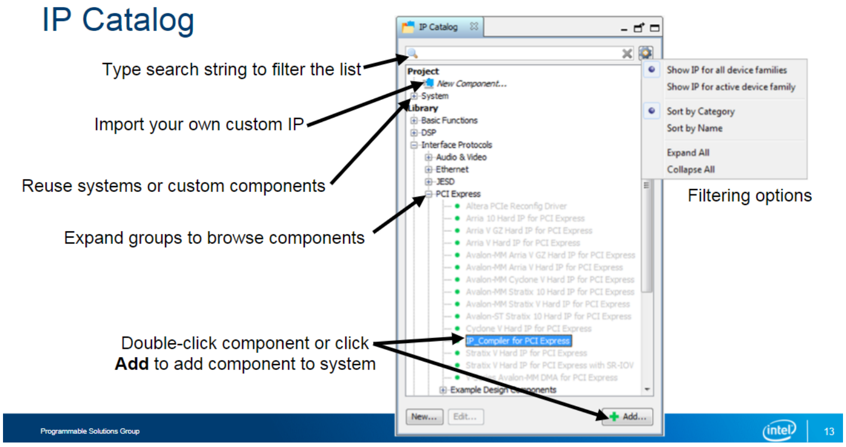
System building flow:

1. Component Selection and Configuration:
   - Begin by choosing components from the IP Catalog for your system.
   - Utilize each component's parameter editor to configure optional settings graphically.
2. Interface Connection in Platform Designer UI:
   - Use the Platform Designer UI to interconnect interfaces of selected components.
   - Remember that a component interface adheres to a standard, comprising one or more signals.
   - Ensure that only interfaces of the same or compatible types are connected.
3. Validation of Connections:
   - Review the established connections to verify their correctness.
   - All necessary connections between components must be accurate to implement the intended data flow in your design.
4. HDL Code Generation and System Integration:
   - Finally, generate HDL code files representing your system.
   - Add the system to an Intel Quartus Prime software project for compilation.

And now we’ll look directly at each one of the steps.

6.6. Select and Configurate components from IP Catalog

IP Catalog



IP Catalog

Type search string to filter the list

Import your own custom IP

Reuse systems or custom components

Expand groups to browse components

Double-click component or click **Add** to add component to system

Filtering options

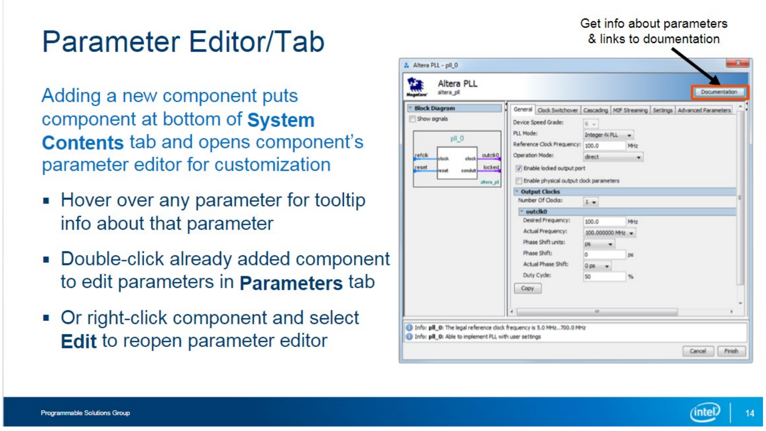
IP Catalog
Project
New Component...
System
Library
Basic Functions
DSP
Interface Protocols
Audio & Video
Ethernet
JESD
PCI Express
Altera PCIe Reconfig Driver
Arria 10 Hard IP for PCI Express
Arria V GZ Hard IP for PCI Express
Arria V Hard IP for PCI Express
Avalon-MM Arria V GZ Hard IP for PCI Express
Avalon-MM Arria V Hard IP for PCI Express
Avalon-MM Cyclone V Hard IP for PCI Express
Avalon-MM Stratix 10 Hard IP for PCI Express
Avalon-MM Stratix V Hard IP for PCI Express
Avalon-ST Stratix 10 Hard IP for PCI Express
Cyclone V Hard IP for PCI Express
IP_Compiler for PCI Express
Stratix V Hard IP for PCI Express
Stratix V Hard IP for PCI Express with SR-IOV
V-Series Avalon-MM DMA for PCI Express
Example Design Components
New...
Edit...
Add...
Show IP for all device families
Show IP for active device family
Sort by Category
Sort by Name
Expand All
Collapse All
Programmable Solutions Group
intel
13

Figure 6 – IP Catalog components

Let’s start from Select and Configurate components from IP Catalog:

1. IP Catalog Overview:
   - The IP Catalog houses available IP for integration into your current Platform Designer system.
   - Unlike the Intel Quartus Prime software's IP Catalog, manual addition to a project is not required for Platform Designer.
2. Categorization of IP:
   - Standard, ready-to-use IP is categorized under the Library section.
   - Custom IP and subsystems are found in the Project section.
   - Custom IP creation details are covered in a subsequent training session linked at the end.
3. Organization for Accessibility:
   - The Library section in the IP Catalog is structured into categories for ease of navigation.
   - Users can expand categories, subcategories, and select components for system inclusion.
   - The search field facilitates quick component location by name, dynamically filtering the list as characters are typed.
4. Search and Compatibility Features:
   - The search field enables users to filter the IP Catalog by name, aiding efficient component identification.
   - Users can narrow down the displayed IP to those compatible with their targeted FPGA device.
5. Adding Components to the System:
   - To add a component into the system, double-click its name or select it and click Add.
   - The chosen component appears in the System Contents tab, and its parameter editor opens for configuration.

Parameter Editor/Tab



Parameter Editor/Tab

Adding a new component puts component at bottom of **System Contents** tab and opens component's parameter editor for customization

- Hover over any parameter for tooltip info about that parameter
- Double-click already added component to edit parameters in **Parameters** tab
- Or right-click component and select **Edit** to reopen parameter editor

Get info about parameters & links to doumentation

Altera PLL - pll_0

Altera PLL
altera_pll

Device Speed Grade: 6
PLL Mode: Integer-N PLL
Reference Clock Frequency: 100.0 MHz
Operation Mode: direct
☑ Enable locked output port
☐ Enable physical output clock parameters

Output Clocks
Number Of Clocks: 1

outclk0
Desired Frequency: 100.0 MHz
Actual Frequency: 100.000000 MHz
Phase Shift units: ps
Phase Shift: 0 ps
Actual Phase Shift: 0 ps
Duty Cycle: 50 %

Info: **pll_0**: The legal reference clock frequency is 5.0 MHz..700.0 MHz
Info: **pll_0**: Able to implement PLL with user settings

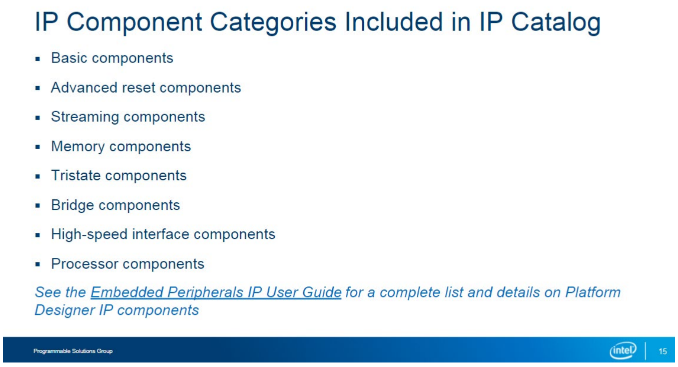
Programmable Solutions Group

14

Figure 7 – Parameter Editor/Tab

1. Adding Components:
    - Double-click or use "Add" to bring a component into the System Contents tab.
    - Opens a window to tweak settings for your system.
2. Individual Setup:
    - Configure each component separately in the parameter editor.
3. Parameter Editor Features:
    - Hover over settings for info.
    - Click "Documentation" for detailed explanations.
4. Compatibility Check:
    - Editor prevents incompatible choices.
    - Check the messages window for any issues.
5. Finalizing Configuration:
    - Click "Finish" to complete setup.
    - Added component now visible in the system.
6. Modification Options:
    - Double-click a component to change settings.
    - Right-click and choose "Edit" for the original editor window.

IP Component Categories Included in IP Catalog



IP Component Categories Included in IP Catalog

- Basic components
- Advanced reset components
- Streaming components
- Memory components
- Tristate components
- Bridge components
- High-speed interface components
- Processor components

See the Embedded Peripherals IP User Guide for a complete list and details on Platform Designer IP components

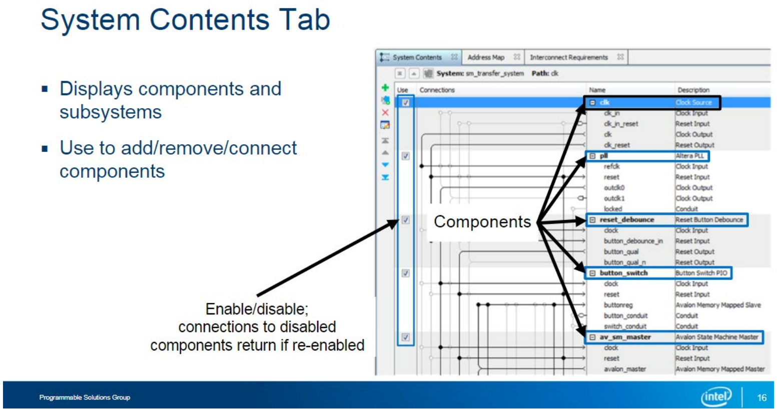
Programmable Solutions Group

intel 15

Figure 8 – IP Component Categories

1. Basic Components:
   - Manage clock and reset signals with Clock Source and Reset Source components.
2. Advanced Reset Components
   - Handle multiple reset signals and gain more control over system reset.
3. Streaming Components:
   - Connect and manipulate data flows between Avalon-ST interfaces.
4. Memory Components:
   - On-chip RAM, ROM, FIFOs, and controllers for off-chip memory (flash, SRAM, SDRAM).
   - On-chip DMA components, including regular and scatter-gather.
5. Tristate Components:
   - Connect to off-chip devices using bi-directional data busses.
   - Share FPGA pins with common signaling for these devices.
6. Bridge Components:
   - Adjust Platform Designer interconnect topology with pipeline stages, asynchronous clock crossing logic, and logic for 32-bit to 64-bit systems.
   - Bridge to external interfaces like JTAG and SPI.
7. High-speed Interface Components:
   - Implement protocol layers for high-speed interfaces such as Ethernet, PCIe*, and Interlaken.
8. Processor Components:
   - Add processors like Nios II soft processor and Arm Cortex processors to your system.
   - Refer to the Embedded Peripherals IP User Guide for a more comprehensive list of available IP components.

System Components Tab



System Contents Tab

- Displays components and subsystems
- Use to add/remove/connect components

Enable/disable; connections to disabled components return if re-enabled

Components

Name	Description
⊟ clk	Clock Source
clk_in	Clock Input
clk_in_reset	Reset Input
clk	Clock Output
clk_reset	Reset Output
⊟ pll	Altera PLL
refclk	Clock Input
reset	Reset Input
outclk0	Clock Output
outclk1	Clock Output
locked	Conduit
⊟ reset_debounce	Reset Button Debounce
clock	Clock Input
button_debounce_in	Reset Input
button_qual	Reset Output
button_qual_n	Reset Output
⊟ button_switch	Button Switch PIO
clock	Clock Input
reset	Reset Input
buttonreg	Avalon Memory Mapped Slave
button_conduit	Conduit
switch_conduit	Conduit
⊟ av_sm_master	Avalon State Machine Master
clock	Clock Input
reset	Reset Input
avalon_master	Avalon Memory Mapped Master

Programmable Solutions Group

intel | 16

Figure 9 – System components tab

System Contents Tab Overview:

Displays all added components and subsystems in the current system.

- Function: Primary interface for system management and configuration.
- Possible actions:
  - Modify a component's relationships within the system.
  - Manage connections to other components and external links.
  - Define a component's location in the memory-mapped master interface.

Example System:

- Components like clock source, PLL, custom button debounce, parallel I/O, and custom state machine are shown.
- Use Column Feature:
  - Enables individual component enable/disable without removing them from the system.

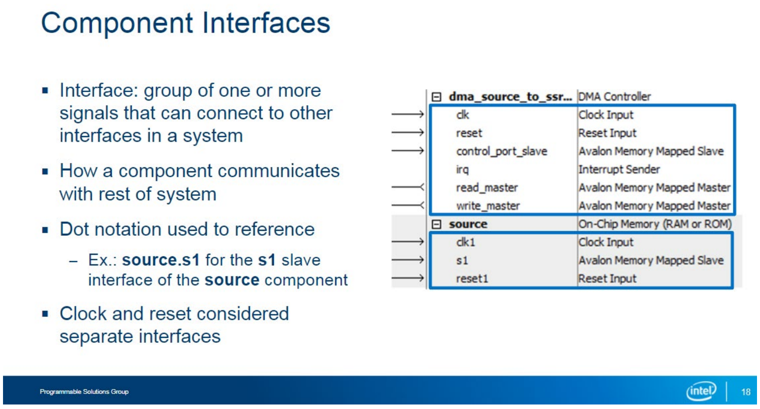
Naming Components:

- Components get default names upon addition.
- Right-click to rename for clarity and meaning.
- Example names: clk, pll, reset_debounce, button_switch, av_sm_master.

Unique Naming:

- Unique names like clk, pll help differentiate multiple instances of the same component.
- Aids organization within the system.

6.7. Component Interfaces



Component Interfaces

- Interface: group of one or more signals that can connect to other interfaces in a system
- How a component communicates with rest of system
- Dot notation used to reference
  - Ex.: **source.s1** for the **s1** slave interface of the **source** component
- Clock and reset considered separate interfaces

⊟ **dma_source_to_ssr...**	DMA Controller
clk	Clock Input
reset	Reset Input
control_port_slave	Avalon Memory Mapped Slave
irq	Interrupt Sender
read_master	Avalon Memory Mapped Master
write_master	Avalon Memory Mapped Master
⊟ **source**	On-Chip Memory (RAM or ROM)
clk1	Clock Input
s1	Avalon Memory Mapped Slave
reset1	Reset Input

Programmable Solutions Group

18

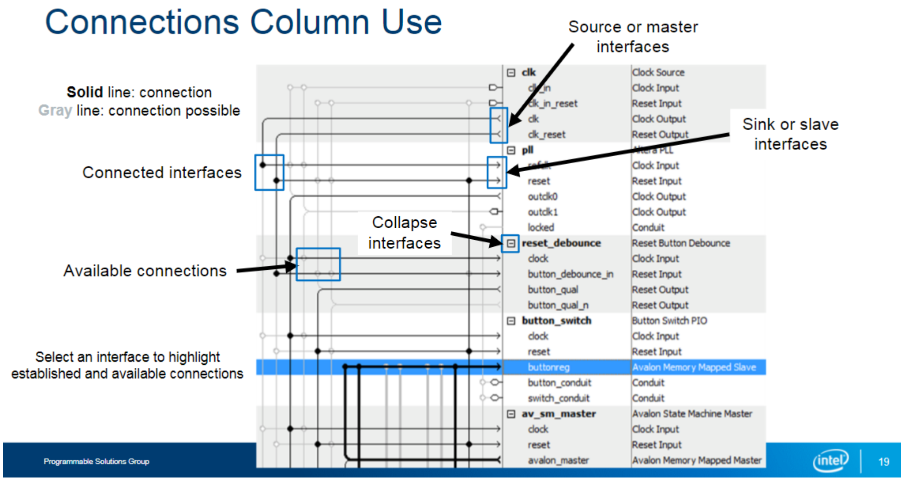
Figure 10 – Component Interfaces

Next one step in the Basic Platform Designer system building flow is Component Interfaces. It allows to connect the component interfaces to each other.

When you add a component to your system, you can see all its communication points, called interfaces, in the System Contents tab. Each interface is like a group of signals that allows the component to talk to other parts of the system. While most interfaces need to be connected, some can be left unconnected, depending on the component. The notation for interfaces uses dot notation, like `source.s1`, where `s1` is an interface of the `source` component.

Unlike component names, you can't change interface names in the Standard edition. However, the Pro edition allows this feature. Clocks and resets are important types of interfaces. Every component has a clock interface that must be connected to a clock source. For resets, they should be linked to a reset source and are synchronous to the component's clock. Keep in mind that Platform Designer doesn't support asynchronous resets. If you leave a reset unconnected, the tool will display a warning, but depending on your design, it might still work.

6.8. Connections Column Use



Connections Column Use

Solid line: connection
Gray line: connection possible

Connected interfaces

Available connections

Select an interface to highlight
established and available connections

Collapse interfaces

Source or master interfaces

Sink or slave interfaces

⊟ **clk**	Clock Source
clk_in	Clock Input
clk_in_reset	Reset Input
clk	Clock Output
clk_reset	Reset Output
⊟ **pll**	Altera PLL
refclk	Clock Input
reset	Reset Input
outclk0	Clock Output
outclk1	Clock Output
locked	Conduit
⊟ **reset_debounce**	Reset Button Debounce
clock	Clock Input
button_debounce_in	Reset Input
button_qual	Reset Output
button_qual_n	Reset Output
⊟ **button_switch**	Button Switch PIO
clock	Clock Input
reset	Reset Input
buttonreg	Avalon Memory Mapped Slave
button_conduit	Conduit
switch_conduit	Conduit
⊟ **av_sm_master**	Avalon State Machine Master
clock	Clock Input
reset	Reset Input
avalon_master	Avalon Memory Mapped Master

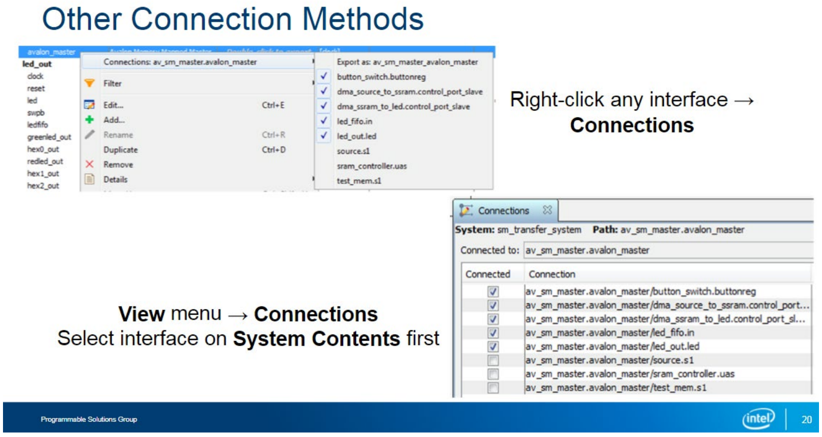
Figure 11 – Usage of the Connection Column

The Connections column in the System Contents tab is a useful tool for understanding how different interfaces of components are linked in your system. It visually represents connections, showing which interfaces are connected and how they connect to one another.

Possible usage:

1. Types of Connections:
   - Quickly identify and make connections like masters to slaves, clock outputs to clock inputs, reset outputs to reset inputs, and more.
2. Graphical Representation:
   - A black dot signifies a connection between two interfaces.
   - An open gray dot indicates interfaces that are not connected but can relate to a simple click.
3. Automatic Generation of Interconnect:
   - Based on parameters in the editor and connections enabled in the column, Platform Designer creates a custom-built interconnect automatically.
4. Additional Details:
   - Arrows show the direction of interfaces - away for masters, clock outputs, reset outputs, sources, or senders, and towards for slaves, clock inputs, reset inputs, sinks, or receivers.
5. Highlighting Connections:
   - Clicking on an interface highlights the connection lines, with black indicating existing connections and gray showing possible connections.
6. Managing Busyness:
   - With many components and connections, the column can get busy.
   - Hide connections by collapsing a component's interfaces - click the + sign next to a component to simplify the display.
7. Interface Filtering Options:
   - Further manage connections and interfaces through filtering options in the System Contents tab and other accessible tabs from the View menu.

6.9. Other Connection Methods



Other Connection Methods

Right-click any interface → **Connections**

View menu → **Connections**
Select interface on **System Contents** first

System: sm_transfer_system **Path:** av_sm_master.avalon_master

Connected to: av_sm_master.avalon_master

Connected	Connection
☑	av_sm_master.avalon_master/button_switch.buttonreg
☑	av_sm_master.avalon_master/dma_source_to_ssram.control_port...
☑	av_sm_master.avalon_master/dma_ssram_to_led.control_port_sl...
☑	av_sm_master.avalon_master/led_fifo.in
☑	av_sm_master.avalon_master/led_out.led
☐	av_sm_master.avalon_master/source.s1
☐	av_sm_master.avalon_master/sram_controller.uas
☐	av_sm_master.avalon_master/test_mem.s1

Figure 12 – Alternative connection methods

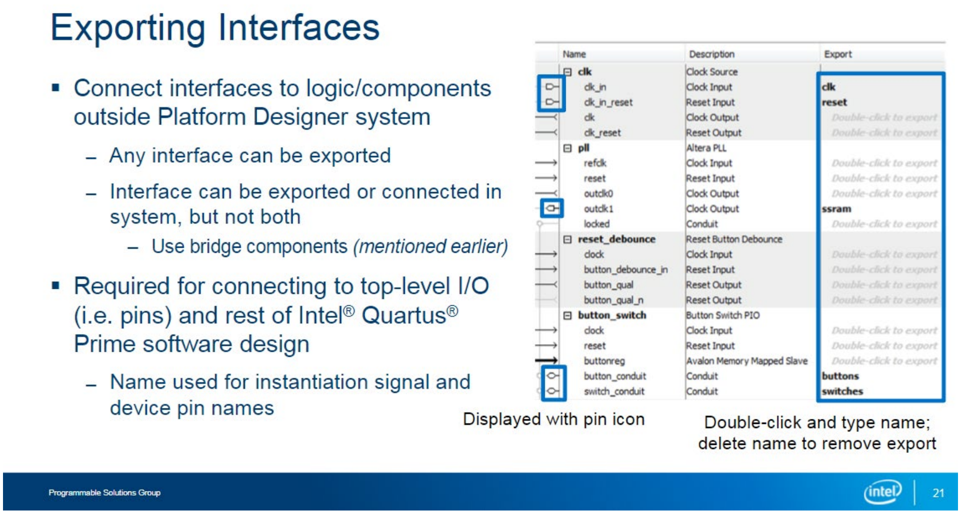
When you add a new clock, master, or streaming sink interface in the System Contents tab, a new line is created in the Connections column to connect other interfaces to it. However, in complex systems, this column can become confusing, leading to potential incorrect connections.

Due to the complexity, use alternate methods to ensure correct connections:

1. Right-Click Method:
   - Right-click any interface and navigate to the Connections sub-menu.
   - It lists all valid connections for the selected interface, with checkmarks indicating existing connections.
   - Click an interface in the list to establish or break its connection with the selected interface.
2. Connections Tab:
   - Accessible from the View menu, the Connections tab offers another method.
   - Select an interface on the System Contents tab, and the Connections tab displays all potential connections for that interface.
   - A slash (/) indicates a possible connection between the selected interface and other compatible interfaces in the system.
   - Use the Connected checkboxes to enable or disable connections.

These alternative methods help you make and verify connections more accurately, especially in intricate system setups.

6.10. Exporting Interfaces



Exporting Interfaces

- Connect interfaces to logic/components outside Platform Designer system
  - Any interface can be exported
  - Interface can be exported or connected in system, but not both
    - Use bridge components *(mentioned earlier)*
- Required for connecting to top-level I/O (i.e. pins) and rest of Intel® Quartus® Prime software design
  - Name used for instantiation signal and device pin names

Name	Description	Export
⊟ **clk**	Clock Source	
clk_in	Clock Input	**clk**
clk_in_reset	Reset Input	**reset**
clk	Clock Output	*Double-click to export*
clk_reset	Reset Output	*Double-click to export*
⊟ **pll**	Altera PLL	
refclk	Clock Input	*Double-click to export*
reset	Reset Input	*Double-click to export*
outclk0	Clock Output	*Double-click to export*
outclk1	Clock Output	**ssram**
locked	Conduit	*Double-click to export*
⊟ **reset_debounce**	Reset Button Debounce	
clock	Clock Input	*Double-click to export*
button_debounce_in	Reset Input	*Double-click to export*
button_qual	Reset Output	*Double-click to export*
button_qual_n	Reset Output	*Double-click to export*
⊟ **button_switch**	Button Switch PIO	
clock	Clock Input	*Double-click to export*
reset	Reset Input	*Double-click to export*
buttonreg	Avalon Memory Mapped Slave	*Double-click to export*
button_conduit	Conduit	**buttons**
switch_conduit	Conduit	**switches**

Displayed with pin icon

Double-click and type name; delete name to remove export

Programmable Solutions Group

intel 21

Figure 13 - Exporting Interfaces

1. What's Exporting?
   - Exporting means taking a component's communication lines (signals) to the top level of Platform Designer.
   - You can connect these signals outside, either to other parts of the Intel Quartus Prime project or directly to the device's I/O pins.
2. How to Export:
   - To export, go to the System Contents tab and double-click in the Export column for the interface you want to export.
   - A default name shows up, but you can change it if you'd like.
   - This name is essential for setting up the system design or making connections to the device's pins.
3. Indicator of Export:
   - After exporting, the visual in the Connections column changes from arrows to a pin icon.
   - This icon shows that the signals from this interface are either coming into or going out of the system.
4. Undoing Export:
   - If you want to bring the exported signals back inside the system, just delete the name in the Export column.
5. Rules about Usage:
   - Remember, an interface can't be both inside and outside the system simultaneously.
   - But, you can get around this by using a bridge component, which duplicates the interface for both internal and external use.
6. Where to Learn More:
   - For extra details, check out the Embedded Peripherals IP User Guide and other resources listed at the end of this training.