Importing HDL Files with Platform Manager 2

August 2014 Technical Note TN1287

Introduction

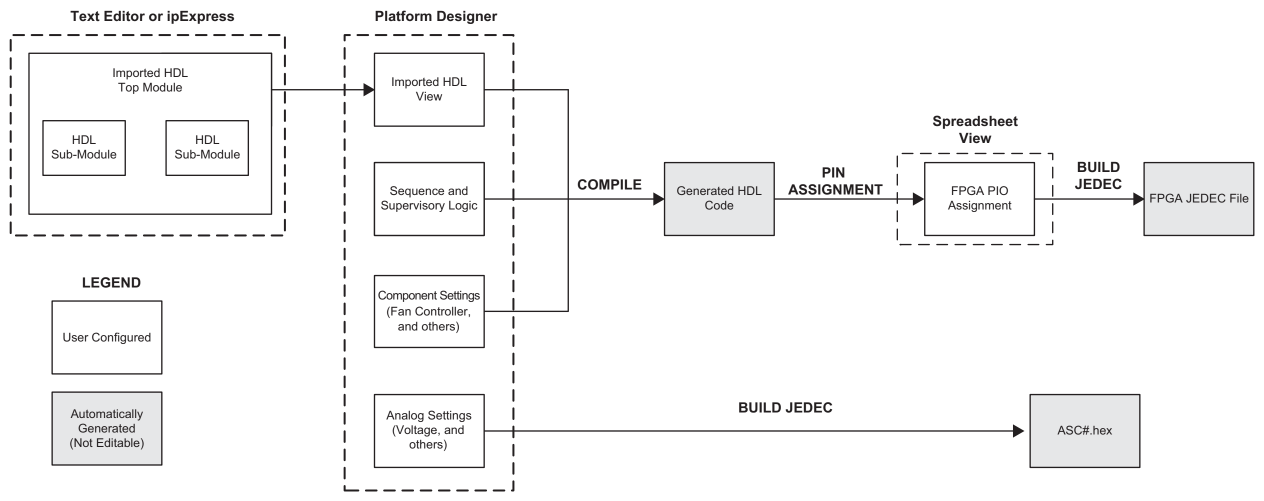
The Platform Manager™ 2 is a fast-reacting, programmable logic based hardware management controller. Platform Manager 2 is an integrated solution combining analog sense and control elements with scalable programmable logic resources. This integrated approach allows the Platform Manager 2 to address Power Management (Power Sequencing, Voltage Monitoring, Trimming and Margining), Thermal Management (Temperature Monitoring, Fan Control, Power Control), and Control Plane functions (System Configuration, I/O Expansion, Reset Distribution, and so forth) as a single device. The hardware management is configured in the Platform Designer software tool (a part of Lattice Diamond® software). This technical note focuses on the Imported HDL feature of Platform Designer. The information in this technical note also applies to hardware management controllers based on MachXO2 combined with one or more L-ASC10 (ASC) hardware management expanders.

Overview

Platform Designer provides the interface for configuring the Power Management, Thermal Management, and Control Plane Functions of Platform Manager 2. Platform Designer includes spreadsheet and graphical interfaces for configuring analog parameters (current monitor settings, voltage monitor settings, temperature monitor settings), system components (fan controller, fault logger, hot swap), and control functions (ports & nodes, logic). The logic design can be implemented using the integrated sequence and supervisory equation builder, or using imported HDL code, or using each tool for a portion of the design.

Connecting the analog sense and control signals to a Verilog or VHDL based logic design is accomplished using the Imported HDL feature in Platform Designer. This feature is used to map HDL ports to signals in Platform Designer and I/O signals in Platform Manager 2. The Imported HDL feature also supports parameter definition in Platform Designer. Figure 1 shows the design flow using the different editors and tool views in Platform Designer and Diamond to generate configuration files for Platform Manager 2. The different tools are highlighted by the dashed lines, with the design steps shown in the different boxes. The white boxes are user configured while the gray boxes are automatically generated by the tool. The Platform Designer build steps are shown above the arrows. This document focuses specifically on the import and connection process related to HDL files.

Figure 1. Platform Designer - Design Flow



Working with Imported HDL Logic

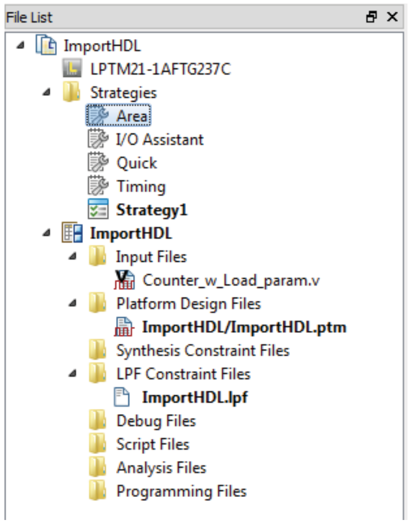
There are three primary steps to importing HDL files into a project. These steps are listed below and described in detail in the following sections. HDL files which will be imported into the project should be edited, synthesized, and simulated prior to being included in the project.

1. Adding HDL Source Files
2. Enabling Imported HDL and Defining Parameters
3. Connecting Imported HDL Modules

Adding HDL Source Files

The input file list is shown in Figure 2. Source files need to be added into this list before they can be imported into Platform Designer. Right-clicking the Input Files folder will bring up a file menu. The Add > Existing File option can be used to add source files to the project.

Figure 2. Input Files - HDL Source



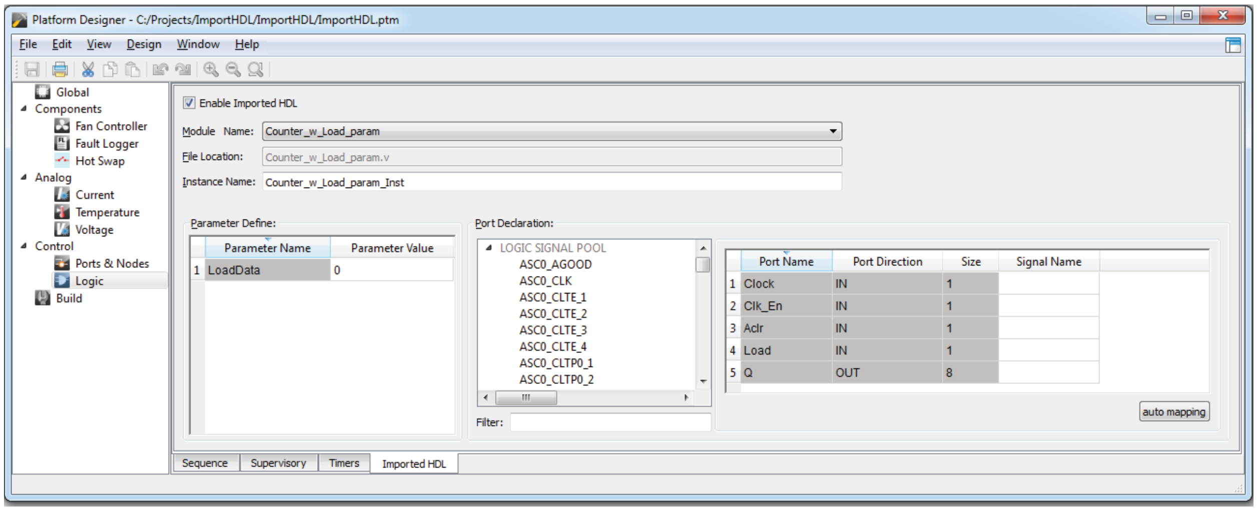
```
File List
ImportHDL
    LPTM21-1AFTG237C
    Strategies
        Area
        I/O Assistant
        Quick
        Timing
        Strategy1
    ImportHDL
        Input Files
            Counter_w_Load_param.v
        Platform Design Files
            ImportHDL/ImportHDL.ptm
        Synthesis Constraint Files
        LPF Constraint Files
            ImportHDL.lpf
        Debug Files
        Script Files
        Analysis Files
        Programming Files
```

Figure 2 shows two example source files – Counter_w_Load.v (Counter module generated by ipExpress, with the Load option enabled) and Counter_w_Load_param.v (a module which replaces the LData input with a parameter and calls the Counter_w_Load module). The imported HDL feature can only import a single module into the Platform Designer project. This single module can be a higher level module which calls multiple sub-modules, as shown in Figure 1. In this way, multiple HDL files can be imported into the project. The imported HDL files may be either Verilog or VHDL source files.

Enabling HDL and Defining Parameters

The Logic view in Platform Designer includes the Imported HDL tab. This tab is used to configure and connect an imported HDL module, as shown in Figure 3.

Figure 3. Imported HDL Tab in Logic View



☑ Enable Imported HDL

Module Name: Counter_w_Load_param

File Location: Counter_w_Load_param.v

Instance Name: Counter_w_Load_param_Inst

Parameter Define:

	Parameter Name	Parameter Value
1	LoadData	0

Port Declaration:

LOGIC SIGNAL POOL
ASC0_AGOOD
ASC0_CLK
ASC0_CLTE_1
ASC0_CLTE_2
ASC0_CLTE_3
ASC0_CLTE_4
ASC0_CLTP0_1
ASC0_CLTP0_2

	Port Name	Port Direction	Size	Signal Name
1	Clock	IN	1	
2	Clk_En	IN	1	
3	Aclr	IN	1	
4	Load	IN	1	
5	Q	OUT	8	

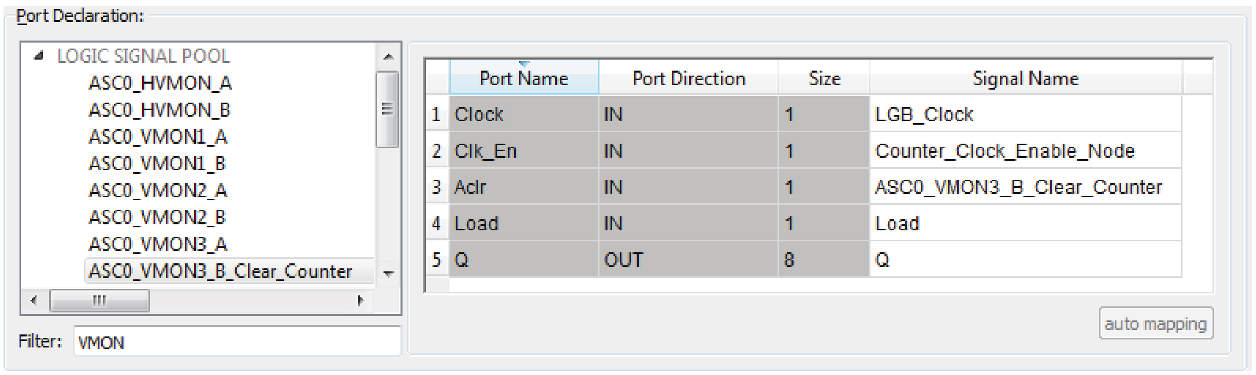
This tab includes several dialog entry boxes, as described in the following list:

- **Enable Imported HDL** – This check box is used to enable the Imported HDL feature. When this box is checked, Platform Designer will parse the HDL files in the Input Files area of the Diamond file list (as shown in Figure 2). Checking the box will return an error if there are no input files in the file list, or if there are syntax errors in files in the input list. Syntax errors will be described in the HDL_parser.log file, as noted by the Error output window.
- **Module Name** – The module name drop-down list will be populated by the top level module found in the input files list. If multiple top level modules are found in the input files list, multiple choices will be available in the module name choices. Figure 3 shows *Counter_w_Load_param* as the chosen top level imported module.
- **File Location** – This is a read only text box, which is automatically populated with the file location of the chosen module from the **Module Name** selection.
- **Instance Name** – This text box is used to name the instance of the module which will be included in the Platform Designer top module which will be generated automatically by the tool. The default name is Module_Name_Inst (where Module_Name is the name chosen in the Module Name box).
- **Parameter Definition** – The Imported HDL window provides an interface for setting parameters if they are available in the imported module. The tool will populate the Parameter Definition section when the module is chosen in the Module Name list. The **Parameter Name** is a read only value defined by the HDL code. The **Parameter Value** will populate with the default parameter values found in the HDL file. The Parameter Value can be set to any legal value in this screen. Figure 3 shows *LoadData* as the parameter name, and *0* as the default parameter value. The imported HDL tab supports parameter definition of numerical constant values only. Values may be entered in binary, decimal, or hexadecimal format.
- **Port Declaration** – The port declaration section is used to assign signals from the Platform Designer signal pool to the imported top module input / output connections. The Logic Signal Pool area is populated with the ASC signals (such as monitor alarms and output controls), any user generated nodes, and the FPGA PIO ports. The **Port Name**, **Port Direction**, and **Size** are all read only values defined by the HDL code. The **Signal Name** is a drag and drop interface used to assign signals from the logic signal pool to the input and output connections of the imported HDL module. The connection process is described in more detail in the next section.
- **Auto Mapping** – The auto mapping button is an alternate method for assigning connections from the signal pool to the imported top module. This button will automatically rename PIO ports (shown in the Ports tab of Ports & Nodes view) to match the imported top module port names and sizes. These updated PIO ports will then be assigned to the associated ports in the Imported HDL tab. This method can be used along with manual signal mapping, as described in the next section.

Connecting Imported HDL Modules

The port declaration section, as previously described, is used to connect the inputs and outputs of the HDL module to signals from the logic signal pool.

Figure 4. Imported HDL Module Port Declaration



Port Declaration:

LOGIC SIGNAL POOL
ASC0_HVMON_A
ASC0_HVMON_B
ASC0_VMON1_A
ASC0_VMON1_B
ASC0_VMON2_A
ASC0_VMON2_B
ASC0_VMON3_A
ASC0_VMON3_B_Clear_Counter

Filter: VMON

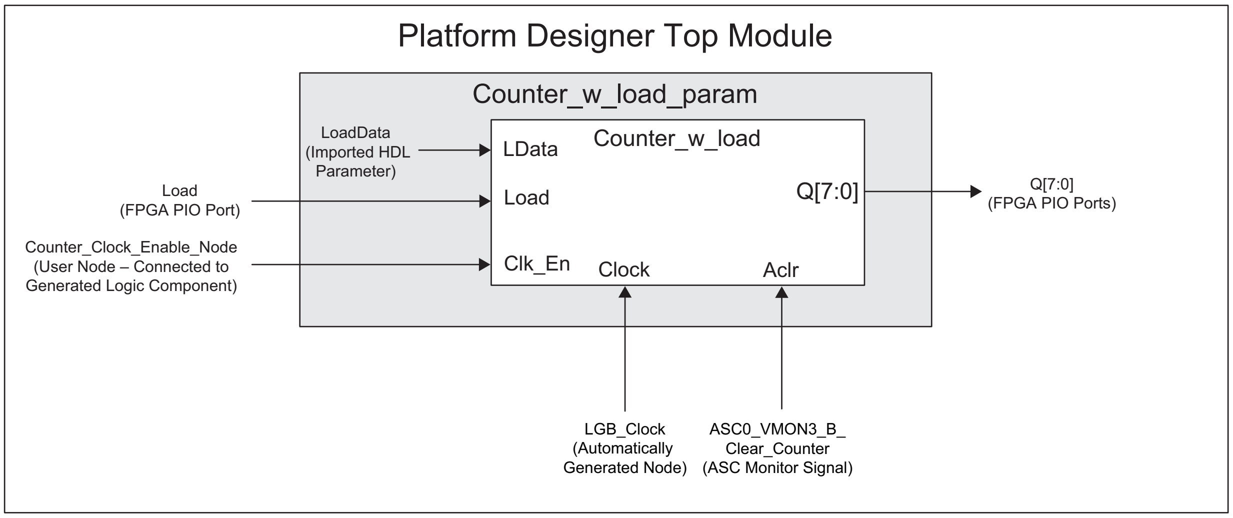
	Port Name	Port Direction	Size	Signal Name
1	Clock	IN	1	LGB_Clock
2	Clk_En	IN	1	Counter_Clock_Enable_Node
3	Aclr	IN	1	ASC0_VMON3_B_Clear_Counter
4	Load	IN	1	Load
5	Q	OUT	8	Q

Figure 4 shows a set of example connections to the Counter_w_Load_Param module. This example shows the different signal pool connection sources. Each of the connections is explained below.

- **Clock** – The clock input signal is tied to the *LGB_Clock*. This is the clock signal used by the Sequence and Supervisory logic. This is an example of connecting to an **automatically generated signal node** in Platform Designer.
- **Clk_En** – This input signal is connected to *Counter_Clock_Enable_Node*. This is a **user generated node** (added to the project in the Ports and Nodes view, Nodes tab). This type of node can be used to connect the generated components (like the Logic Sequence or a Fan Controller) to the Imported HDL module.
- **Aclr** – This input signal is connected to ASC0_VMON3_B_Clear_Counter. This is an **output monitor signal from the ASC** Voltage Monitor circuits. The Imported HDL module can be connected to the ASC monitor or control signals.
- **Load** – This input signal is connected to an **input port** called Load. This connection was generated automatically by clicking the auto mapping button. (The auto mapping feature should be used after any manually assigned nodes or ports have been assigned). The port can be assigned to an FPGA I/O pin in the spreadsheet view during the project build process.
- **Q** – This output group signal is connected to an **8-bit output port** which has been generated automatically using the auto mapping button. These eight outputs can be assigned to FPGA I/O pins in the spreadsheet view, during the project build process.

The final module connection is shown in Figure 5.

Figure 5. Imported HDL - Example Module Connections


Platform Designer Top Module
Counter_w_load_param
Counter_w_load
LoadData (Imported HDL Parameter)
LData
Load (FPGA PIO Port)
Load
Counter_Clock_Enable_Node (User Node – Connected to Generated Logic Component)
Clk_En
Clock
Aclr
Q[7:0]
Q[7:0] (FPGA PIO Ports)
LGB_Clock (Automatically Generated Node)
ASC0_VMON3_B_ Clear_Counter (ASC Monitor Signal)

Summary

The flexibility of the Imported HDL feature in Platform Designer allows hardware management designers to implement their logic according to their preferred design flow. The steps for working with the Imported HDL feature with Platform Manager 2 have been described.

References

- Platform Designer User Guide

Technical Support Assistance

e-mail: techsupport@latticesemi.com

Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
August 2014	1.0	Initial release.