Using the Warp® Development Tool for the Delta39K™ CPLD Family–Advanced Users

Introduction

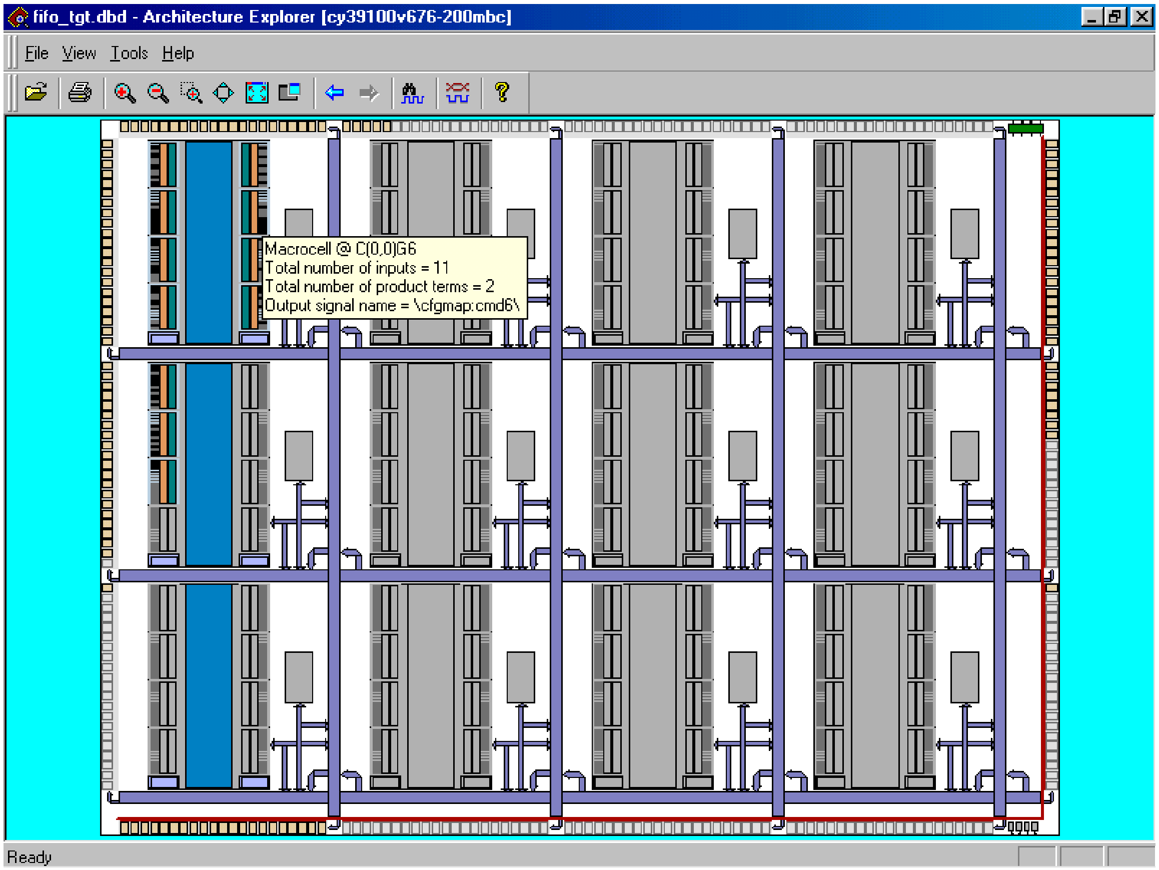
The Delta39K™ CPLD family of programmable logic devices has several more capabilities and features beyond the preceding families, and so required additional tool support to access these features. The *Warp*® Tool is Cypress's main compiler (synthesis and fitting) engine for all Cypress programmable logic products. Galaxy is the executable main tool in the *Warp* Tool Suite, and spawns the following tools. The Architecture Explorer, Timing Analyzer and I/O Properties tools were added to the *Warp* tool for the Delta39K CPLD family. This application note will walk the user through the various features and capabilities of the Architecture Explorer, Timing Analyzer, and I/O Properties tools.

This application note assumes the user is familiar with the *Warp* tool, and that the user has already compiled a design targeting a member of the Delta39K CPLD family. This application note uses a PCI core as an example design.

Architecture Explorer

To access the Architecture Explorer from Galaxy, click on Tools, then Architecture Explorer. (Architecture Explorer can also be accessed through the Windows start button under Programs in the Cypress *Warp* tool directory.)

This brings up a map of the device that was targeted for the design. This map shows utilized resources in color and unused resources in gray. The Delta39K CPLDs use a product-term-based architecture for its basic building blocks. *Figure 1* shows a CY39100. This device has approximately 100,000 FPGA-equivalent gates. It implements this using the twelve clusters shown on this map. Each cluster contains eight Logic Blocks (LB), and each LB contains 16 Macrocells (MC). The LB and MC structure of this family is very much like the Ultra37000™ CPLD family.



Macrocell @ C(0,0)G6
Total number of inputs = 11
Total number of product terms = 2
Output signal name = \cfgmap:cmd6\

Figure 1. CY39100 in the *Warp* Tool's Architecture Explorer

Identifying Blocks Within the Architecture

Individual clusters are identified using the following label format: C(<row>,<column>) where the rows and columns are numbered from the top left, starting with 0. For example, the cluster third from the left and second down would be labeled “C(1,2)”.

Individual LBs are identified by letters, starting with A in the upper left corner, counting around the cluster in a “U” shape using the following label format: C(<row>,<column>)<LB>. For example, the logic block in the lower right corner of the cluster discussed above would be labeled “C(1,2)E”.

Individual MCs are identified using the following label format: C(<row>,<column>)<LB><MC> where the MCs are numbered from the top of each LB starting with 0. For example, the twelfth macrocell down in the Logic Block discussed above would be labeled “C(1,2)E11”.

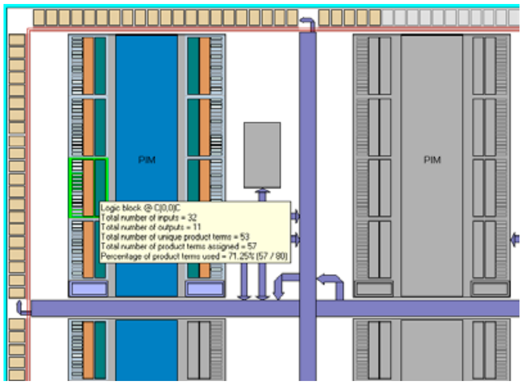
Individual cluster memory blocks are identified using the following label format: C(<row>,<column>)<MB> with block 0 to the right and 1 to the left. Individual channel memory blocks are identified using the same label format as their adjacent clusters.

Individual I/O groups are identified using the following label format: <edge><I/O Block><I/O Cell> where the first letter indicates Top, Bottom, Right, or Left, and the second letter is assigned in order from left to right and from top to bottom. In each group, individual I/O cells are identified by adding a number at the end of the above format, starting with 0, assigned in order from left to right and from top to bottom. For example, the third block down on the left is labeled “LC”, and the fourteenth cell from the left, in the second block from the left, on the top edge is labeled “TB13”.

There are four global control lines. They are shown in the lower right corner. There is one PLL block. It is identified as “Global Clock Cell”. It is shown in the upper right corner. There is only one, so it has no abbreviated label.

View Features

The Architecture Explorer has several convenient features for viewing and analyzing the design layout. *Figure 2* shows that pointing the curser at any functional block will automatically bring up a tool tip window with general signal and usage statistics on that item. Single clicking on that same location will highlight the block in green, as shown in *Figure 3*.



```
Logic block @ C(0,0)C
Total number of inputs = 32
Total number of outputs = 11
Total number of unique product terms = 53
Total number of product terms assigned = 57
Percentage of product terms used = 71.25% (57 / 80)
```

Figure 2. Logic Block Tool Tip

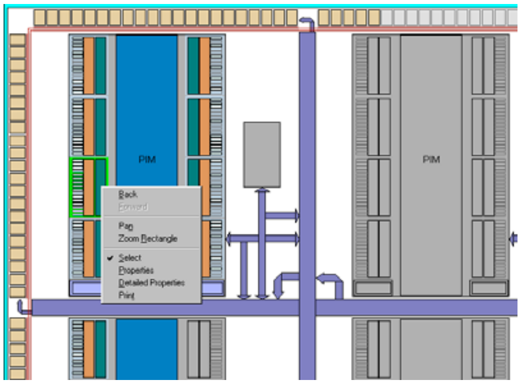
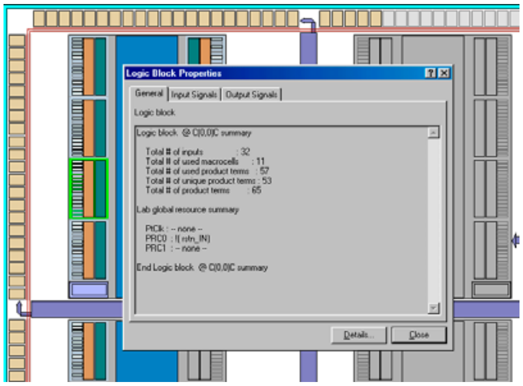


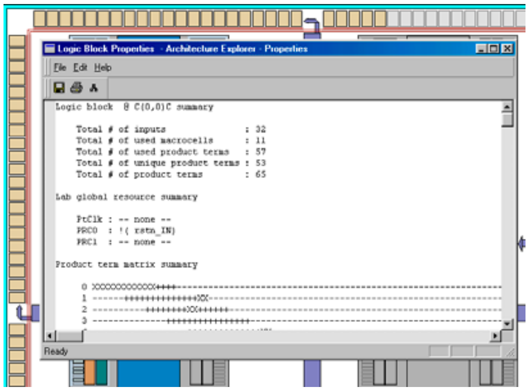
Figure 3. Logic Block Properties

Double clicking on that location will bring up more detailed signal and usage information. This same level of information can be accessed by right-clicking and selecting Properties. This is shown in *Figure 4*. The highest level of detail, shown in *Figure 5*, can be accessed by right clicking and selecting Detailed Properties, or by clicking the Detail button on the Properties window.



```
Logic block  @ C(0,0)C summary

   Total # of inputs              : 32
   Total # of used macrocells     : 11
   Total # of used product terms  : 57
   Total # of unique product terms : 53
   Total # of product terms       : 65

Lab global resource summary

   PtClk : -- none --
   PRC0  : !( rstn_IN)
   PRC1  : -- none --

End Logic block  @ C(0,0)C summary
```

Figure 4. Logic Block Right-Click Menu



```
Logic block  @ C(0,0)C summary

    Total # of inputs               : 32
    Total # of used macrocells      : 11
    Total # of used product terms   : 57
    Total # of unique product terms : 53
    Total # of product terms        : 65

Lab global resource summary

    PtClk : -- none --
    PRC0  : !( rstn_IN)
    PRC1  : -- none --

Product term matrix summary

    0 XXXXXXXXXXXX++++--------------------------------------------------------------
    1 ------++++++++++++++XX--------------------------------------------------------
    2 ----------++++++++XX++++++----------------------------------------------------
    3 --------------++++++++++++++++------------------------------------------------
```

Figure 5. Logic Block Detailed Properties

The Zoom In and Zoom Out buttons on the menu bar will zoom from the center of the window. The Zoom Rectangle button toggles on or off and will allow you to select the exact level of zoom. The Pan option toggles on and off, and allows you to click and drag the cursor to move the map inside the window. The Zoom to Fit button will bring the window back to a global view, and the Overview Window will bring up a thumbnail global view, highlighting the area presently viewed in the main window (see *Figure 6*). These features can also be accessed in the View menu.

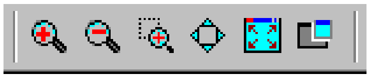


Figure 6. Zoom Buttons

The back and forward arrows behave like those on an Internet browser. The Find Signal button brings up a list of all the signals defined in the design or generated by the compiler to implement the design (see *Figure 7*). When a signal in the list is highlighted, its source and destination are shown on the map.

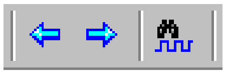


Figure 7. Navigation Buttons

Timing Analyzer

Purpose

The Timing Analyzer is a static timing analysis tool that is an integral part of Architecture Explorer. Together, they provide a wide variety of information about the compiler's placement results, in graphical and tabular formats.

Basic Capabilities

The Timing Analyzer focuses on the worst-case timing results of the signal paths chosen by the compiler. It provides its timing summaries in a spreadsheet or table format, sorted from the longest delay of any kind in the design, to the shortest. The tool provides some tables of pre-filtered subsets of the whole list, grouped according to delay type, like t_{PD}, t_{SU}, t_{CO}, etc. The tool will further allow creation of custom tables using filter criteria provided by the user. The criteria can include things like source, destination, delay type, or signal name.

Design Flow

As a static timing analyzer, this tool will provide delay estimates for every possible signal in the design, even if some of the delays would never occur in the function of the circuit. This will sometimes create more information than is useful, and create the need to filter the list to view relevant signals. Static timing analyzers also do not handle asynchronous feedback well. This tool is only intended for post-synthesis analysis, and is not intended for entering timing requirements as directives to the compiler.

How It Works

The Timing Analyzer can be accessed through the Timing Analyzer button (shown in *Figure 8*) or under the Tools menu.

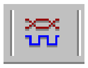
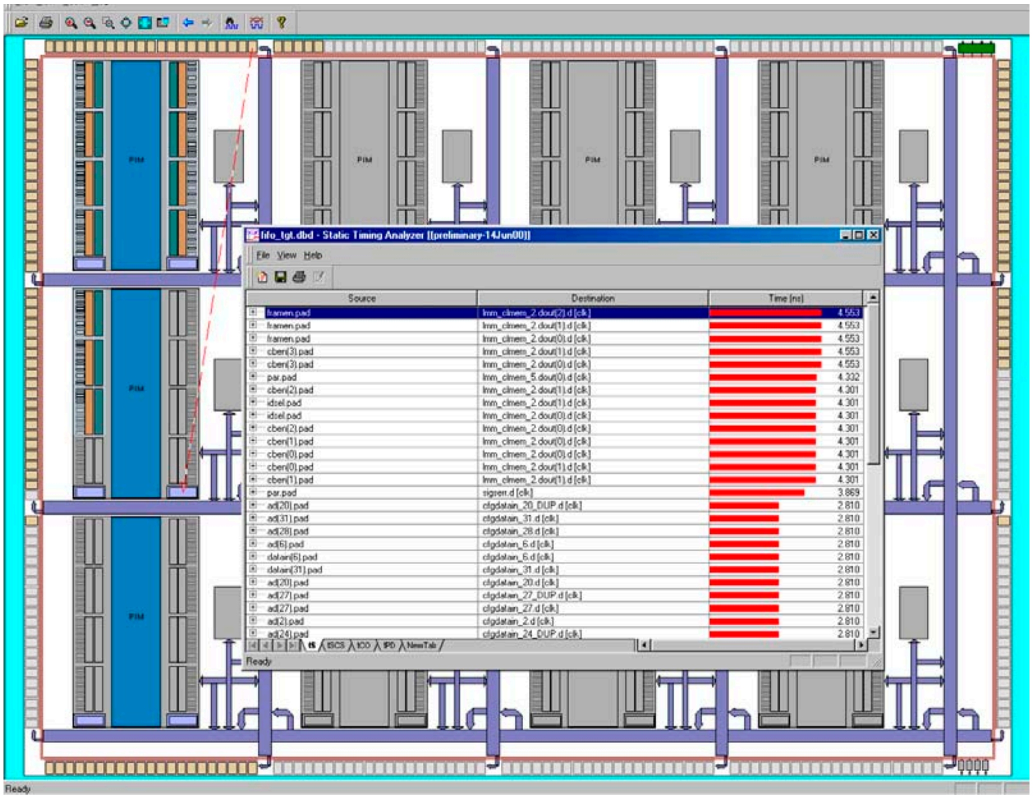


Figure 8. Timing Analyzer Button

The Timing Analyzer window comes up over the Architecture Explorer (*Figure 9*). This tool contains several spreadsheet style lists of signals and timing values including the source and destination of each signal path.



fifo_tgt.dbd - Static Timing Analyzer [[preliminary-14Jun00]]

Source	Destination	Time (ns)
framen.pad	lmm_clmem_2.dout(2).d [clk]	4.553
framen.pad	lmm_clmem_2.dout(1).d [clk]	4.553
framen.pad	lmm_clmem_2.dout(0).d [clk]	4.553
cben(3).pad	lmm_clmem_2.dout(1).d [clk]	4.553
cben(3).pad	lmm_clmem_2.dout(0).d [clk]	4.553
par.pad	lmm_clmem_5.dout(0).d [clk]	4.332
cben(2).pad	lmm_clmem_2.dout(1).d [clk]	4.301
idsel.pad	lmm_clmem_2.dout(1).d [clk]	4.301
idsel.pad	lmm_clmem_2.dout(0).d [clk]	4.301
cben(2).pad	lmm_clmem_2.dout(0).d [clk]	4.301
cben(1).pad	lmm_clmem_2.dout(0).d [clk]	4.301
cben(0).pad	lmm_clmem_2.dout(0).d [clk]	4.301
cben(0).pad	lmm_clmem_2.dout(1).d [clk]	4.301
cben(1).pad	lmm_clmem_2.dout(1).d [clk]	4.301
par.pad	sigserr.d [clk]	3.869
ad(20).pad	cfgdatain_20_DUP.d [clk]	2.810
ad(31).pad	cfgdatain_31.d [clk]	2.810
ad(28).pad	cfgdatain_28.d [clk]	2.810
ad(6).pad	cfgdatain_6.d [clk]	2.810
datain(6).pad	cfgdatain_6.d [clk]	2.810
datain(31).pad	cfgdatain_31.d [clk]	2.810
ad(20).pad	cfgdatain_20.d [clk]	2.810
ad(27).pad	cfgdatain_27_DUP.d [clk]	2.810
ad(27).pad	cfgdatain_27.d [clk]	2.810
ad(2).pad	cfgdatain_2.d [clk]	2.810
ad(24).pad	cfgdatain_24_DUP.d [clk]	2.810

Figure 9. Timing Analyzer Window

The rows are listed in order from longest to shortest delay, and the delay is shown in numeric and bar graph formats. When a row is selected, its signal path is highlighted on the device map. The individual segments of the delay path are detailed on the spreadsheet by clicking the plus box on the left of the row. The tabs at the bottom provide a selection of prefiltered timing parameters, including t_S, t_{SCS}, t_{CO} and t_{PD}.

Timing Models

It is important to understand how the Timing Analyzer determines the delay values. The Timing Analyzer determines the delay values by tracing the path of each individual signal that the compiler routed. It then identifies every intermediate segment along the signal's path. It then refers to a database of worst-case delay values for these segments, called timing models. It finally adds these segment values to provide a worst-case total value for the signal delay. Statistically, this sum-of-worst-cases is worse than any real-world worst-case. This is because it is extremely rare that all the segments along a signal path will actually perform at their worst values simultaneously. This is an unfortunate limitation of timing analyzers used to estimate timing of complex systems–systems whose signal timing is determined from several segments. This limitation could also be viewed as a benefit. While most final values are reported as worse than real-world worst-case values, this limitation builds in timing margin for signals whose reported value is adequate for the design.

Block Delays

The Timing analyzer will provide propagation delay values such as t_{CO}, t_{PD}, and t_{HZ}. It also provides block delay values. A block delay is the signal propagation delay value between registers on the same clock. Therefore, the signal with the largest block delay–delay between registers–represents the minimum possible clock period. This block delay value is called t_{SCS}, and the inverse of largest t_{SCS} (the minimum period) is the maximum frequency for the design. In the Delta39K CPLD architecture, additional delays are accumulated when signals travel from one cluster to another, and the worst delays are accumulated when signals are routed from horizontal to vertical channels outside the clusters. The highest frequencies can be achieved first by implementing sequentially clocked logic within a single cluster, or second by placing the downstream registers in the same cluster column or row.

Additional Filtering

A "New Tab" capability is also included. This new tab is associated with a blank spreadsheet to which custom signal filtering can be added. Additional tabs can be created by clicking the Add Custom Sheet button (*Figure 10*) or by selecting Add Custom Sheet under the View menu. These custom sheets are provided for specialized filtering, and the filtering can be configured in the window accessed through the "Edit Custom Sheet" button (*Figure 10*) or the Edit Custom Sheet selection under the View menu.

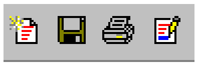
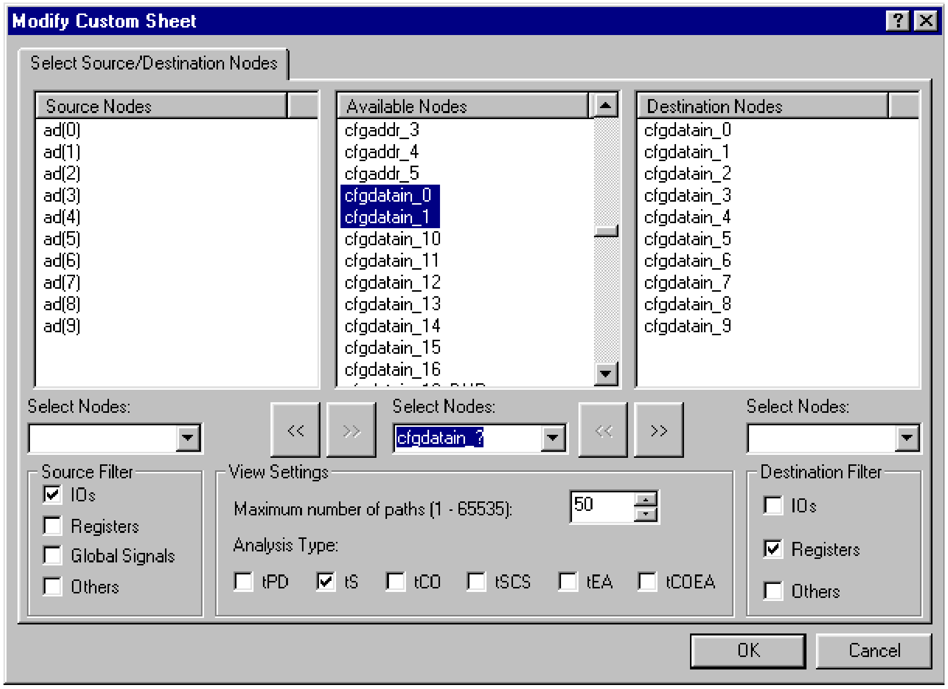


Figure 10. Additional Tabs

In the Modify Custom Sheet window (*Figure 11*) the available signals are listed in the center Available Nodes list. There are several ways to filter the list for the desired signals. First, the source signal names can be selected by checking the source type in the Source Filter area. Second, they can be further selected with the text filter in the Select Nodes box under both Source Nodes and Available Nodes. DOS wildcards can be used in the Select Nodes boxes, or the signals can then be individually selected from the list (using CTRL or Shift to select multiple signals). Once the desired signals have been highlighted, they can be included in or excluded from the list



Modify Custom Sheet

Select Source/Destination Nodes

Source Nodes: ad(0), ad(1), ad(2), ad(3), ad(4), ad(5), ad(6), ad(7), ad(8), ad(9)

Available Nodes: cfgaddr_3, cfgaddr_4, cfgaddr_5, cfgdatain_0, cfgdatain_1, cfgdatain_10, cfgdatain_11, cfgdatain_12, cfgdatain_13, cfgdatain_14, cfgdatain_15, cfgdatain_16

Destination Nodes: cfgdatain_0, cfgdatain_1, cfgdatain_2, cfgdatain_3, cfgdatain_4, cfgdatain_5, cfgdatain_6, cfgdatain_7, cfgdatain_8, cfgdatain_9

Select Nodes:

Select Nodes: cfgdatain_?

Select Nodes:

Source Filter: ☑ IOs ☐ Registers ☐ Global Signals ☐ Others

View Settings

Maximum number of paths (1 - 65535): 50

Analysis Type: ☐ tPD ☑ tS ☐ tCO ☐ tSCS ☐ tEA ☐ tCOEA

Destination Filter: ☐ IOs ☑ Registers ☐ Others

Figure 11. Modify Custom Sheet Window

with the Include and Exclude buttons. A similar procedure applies to the Destination Nodes. Finally, the View settings area is used to determine which parameter is to be viewed. *Figure 11* shows a custom filter designed to display the set-up time for the configuration register.

How to Determine System Performance

Here are some specific procedures for viewing values critical in determining system performance.

- **Setup Time Analysis Procedure**: To duplicate the worksheet with the tab labeled t_S: in the Source Filter box select IOs, Registers, Global Signals and Others; in the View Settings box select t_S (make sure the Maximum number of paths is set high enough); and in the Destination Filter box select Registers. Then highlight all the signals in the Available Nodes list and capture the filtered signals into Source Signals list and then into the Destination Nodes list with the include buttons.
- **Clock to Output Analysis Procedure**: To duplicate the worksheet with the tab labeled t_{CO}: in the Source Filter box select Global Signals; in the View Settings box select t_{CO} (make sure the Maximum number of paths is set high enough); and in the Destination Filter box select IOs. Then highlight all the signals in the Available Nodes list and capture the filtered signals into Source Signals list and then into the Destination Nodes list with the include buttons.
- **f_{MAX} Analysis Procedure**: As discussed in the Block Delays section, t_{SCS} is the propagation delay between registers. The largest t_{SCS} for each clock is the minimum period for that clock, so the inverse of that value is f_{MAX}. To duplicate the worksheet with the tab labeled t_{SCS}: in the Source Filter box select Registers; in the View Settings box select t_{SCS} (make sure the Maximum number of paths is set high enough); and in the Destination Filter box select Registers. Then highlight all the signals in the Available Nodes list and capture the filtered signals into Source Signals list and then into the Destination Nodes list with the include buttons.
- **Asynchronous Propagation Delay Analysis Procedure**: To duplicate the worksheet with the tab labeled t_{PD}: in the Source Filter box select IOs and Global Signals; in the View Settings box select t_{PD} (make sure the Maximum number of paths is set high enough); and in the Destination Filter box select IOs. Then highlight all the signals in the Available Nodes list and capture the filtered signals into Source Signals list and then into the Destination Nodes list with the include buttons.
- **Global Control Input to Output Enable Analysis Procedure**: To create a worksheet to analyze t_{EA}: in the Source Filter box select Global Signals; in the View Settings box select t_{EA} (make sure the Maximum number of paths is set high enough); and in the Destination Filter box select IOs. Then highlight all the signals in the Available Nodes list and capture the filtered signals into Source Signals list and then into the Destination Nodes list with the include buttons.
- **Global Control Register Clock to Output Enable Analysis Procedure**: To create a worksheet to analyze t_{COEA}: in the Source Filter box select Global Signals; in the View Settings box select t_{COEA} (make sure the Maximum number of paths is set high enough); and in the Destination Filter box select IOs. Then highlight all the signals in the Available Nodes list and capture the filtered signals into Source Signals list and then into the Destination Nodes list with the include buttons.
- **Customized Measurements**: To look at a subset of one of these tables, first clear the Source Nodes and the Destination Nodes lists. Categories in the Source Filter or Destination Filter boxes can be deselected (as long as one category in each box remains checked). Or signals can be filtered by name with the text filter in the Select Nodes box under both Source Nodes and Available Nodes as described above. Once the selections in these areas are made, capture the filtered signals into Source Signals list and then into the Destination Nodes list with the include buttons.

Critical Path Analysis

When a signal in one of the tables is selected and highlighted, the Architecture Explorer will display arrows indicating the path of the signal on the device map.

The largest additive delays are in the horizontal to vertical or vertical to horizontal intersections. Placing source and destination nodes of critical signals in the same cluster column or the same cluster row eliminates the use of these intersections and removes the greatest delays.

The second greatest delays are incurred between clusters in the same cluster row or cluster column. Placing source and destination nodes of critical signals in the same cluster will provide the best possible timing results in the Delta39K CPLD architecture.

To further reduce propagation delays within a cluster, the normal rules of CPLD design apply. Since the Delta39K CPLD cluster is based on the architecture of the Ultra37000 CPLD family, the timing within each cluster is very deterministic. Signal delays are not as dependent on numbers of product terms used as they are in other CPLD architectures. So the best timing results within clusters can usually be obtained by allowing the *Warp* tool to flatten and optimize equations and optimize signal placement.

For more information on timing optimization within a cluster, see the “How Is This Useful” notes in the application note entitled *Understanding the Warp® Report File for Ultra37000™ Devices*.

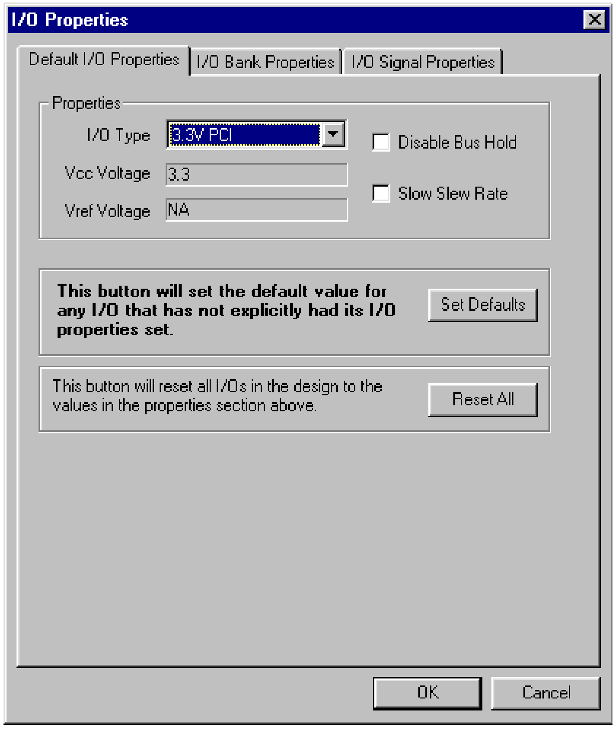
Exporting Timing Information

The information in any spreadsheet can be printed by clicking the printer button (*Figure 10*) or by selecting Print under the File menu. The information in any spreadsheet can also be saved to a test file by selecting Save As under the File menu.

I/O Properties

I/O properties can be accessed from Galaxy by selecting Configure IO Standards under the Project menu. There are three tabs in this window, allowing different levels of control over the I/O pins. They are shown in *Figure 12* through *Figure 14*.

The first tab, shown in *Figure 12*, controls the highest level of settings. The I/O type can be chosen from the I/O Type pull-down menu and use of the bus hold and slew rate features can also be determined. After the desired properties have been assigned, the Set Defaults button will assign those properties to any I/O that has not already been explicitly set. The Reset All button will overwrite all assignments.



I/O Type: 3.3V PCI

Vcc Voltage: 3.3

Vref Voltage: NA

□ Disable Bus Hold

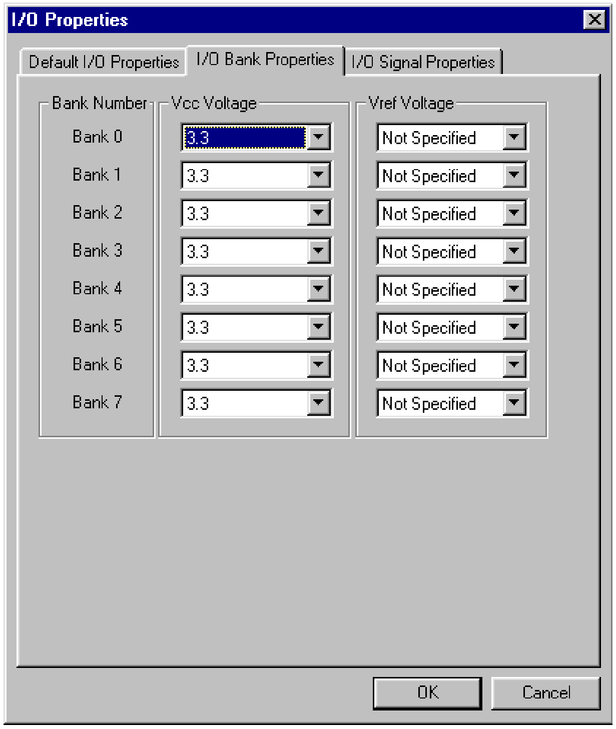
□ Slow Slew Rate

This button will set the default value for any I/O that has not explicitly had its I/O properties set.

This button will reset all I/Os in the design to the values in the properties section above.

Figure 12. I/O Properties GUI (Default)

All Delta39K CPLD devices have eight I/O banks each with independent I/O power supply and reference voltages. The second tab, I/O Bank Properties (shown in *Figure 13*) allows the user to set the V_{CC} and V_{REF} voltages available to each group or bank of I/O pins.

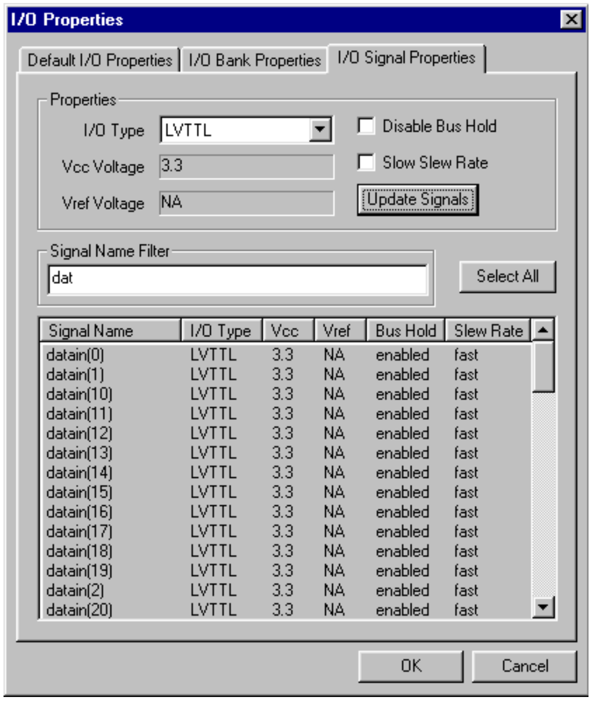


Bank Number	Vcc Voltage	Vref Voltage
Bank 0	3.3	Not Specified
Bank 1	3.3	Not Specified
Bank 2	3.3	Not Specified
Bank 3	3.3	Not Specified
Bank 4	3.3	Not Specified
Bank 5	3.3	Not Specified
Bank 6	3.3	Not Specified
Bank 7	3.3	Not Specified

Figure 13. I/O Properties GUI (Bank)

Once the voltages are applied the standards selected in the I/O Signal Properties tab (*Figure 14*) must match the voltages made available in the I/O Bank Properties tab. I/O standards that share the same V_{REF} and V_{CC} requirements can be assigned to the same I/O block. I/O standards that use only V_{CC} can be assigned to blocks with other standards that share the same requirement, regardless of the V_{REF} assignment. Refer to the application note *Delta39K and Quantum38K™ I/O Standards and Configurations* for further detail.

In the I/O Signal Properties tab, the list of signals can be filtered using the Signal Name Filter box. Wildcards are inferred at the beginning and end of the inserted character string, and DOS wildcards do not apply. The signals can then be individually selected from the list (using CTRL or Shift to select multiple signals), or the entire filtered list can be selected with the Select All button. Then the I/O type is chosen from the I/O Type pull-down menu, and use of the bus hold and slew rate features can also be determined. The new standard is applied to the selected signals when the Update Signal button is pressed.



Properties

I/O Type: LVTTL

Vcc Voltage: 3.3

Vref Voltage: NA

□ Disable Bus Hold

□ Slow Slew Rate

Signal Name Filter: dat

Signal Name	I/O Type	Vcc	Vref	Bus Hold	Slew Rate
datain(0)	LVTTL	3.3	NA	enabled	fast
datain(1)	LVTTL	3.3	NA	enabled	fast
datain(10)	LVTTL	3.3	NA	enabled	fast
datain(11)	LVTTL	3.3	NA	enabled	fast
datain(12)	LVTTL	3.3	NA	enabled	fast
datain(13)	LVTTL	3.3	NA	enabled	fast
datain(14)	LVTTL	3.3	NA	enabled	fast
datain(15)	LVTTL	3.3	NA	enabled	fast
datain(16)	LVTTL	3.3	NA	enabled	fast
datain(17)	LVTTL	3.3	NA	enabled	fast
datain(18)	LVTTL	3.3	NA	enabled	fast
datain(19)	LVTTL	3.3	NA	enabled	fast
datain(2)	LVTTL	3.3	NA	enabled	fast
datain(20)	LVTTL	3.3	NA	enabled	fast

Figure 14. I/O Properties GUI (Signal)

Figure 12 through *Figure 14* illustrate setting all of the I/O to the 3.3V PCI standard, and then selecting and setting the I/O on the user side of this PCI Bridge to LVTTL.

The I/O properties are saved in the project Control file. This file can be viewed and edited by selecting Control File under Galaxy's View menu.

Warp is a registered trademark, and Delta39K, Ultra37000, and Quantum38K are trademarks, of Cypress Semiconductor. All product and company names mentioned in this document are the trademarks of their respective holders.

approved AN019 3/27/03 kkv