CY3130

Warp3® VHDL and Verilog Development System for CPLDs

Features

- **Sophisticated CPLD design and verification system based on VHDL and Verilog**
- ***Warp3*® is based on the Workview Office® (PC) design environment**
  - **—Advanced graphical user interface for Windows®**
  - **—Schematic capture (ViewDraw®)**
  - **—VHDL source-level simulator (SpeedWave®)**
  - **—Interactive timing simulator (ViewSim®)**
  - **—Waveform stimulus and viewing (ViewTrace®)**
  - **—Textual design entry using VHDL and Verilog**
  - **—Mixed-mode (schematics and text) design entry support for VHDL**
- **The core of *Warp3* consists of an IEEE1076 and 1164 standard VHDL and an IEEE 1364 Verilog compiler**
  - **—Open, powerful design languages**
  - **—Facilitate design portability across devices and/or CAD platforms**
  - **—Facilitate the use of industry-standard simulation and synthesis tools for board and system-level design**
- **Additional option for timing simulation and an FSM editor from Aldec**
  - **—Active-HDL™ Sim Release 3.3 graphical waveform simulator**
  - **—Active-HDL FSM graphical Finite State Machine editor**
- **Support for ALL Cypress PLDs and CPLDs including:**
  - **—Ultra37000™ CPLDs (now with FBGA support)**
  - **—FLASH370i™ CPLDs**
  - **—MAX340™ CPLDs**
  - **—Industry-standard PLDs (16v8, 20v8, 22v10)**

Introduction

As the capacity and complexity of programmable logic increased dramatically over the last few years, users began to demand software tools that would allow them to manage this growing complexity. They also began to demand design-entry standards that would allow them to spend more time designing with PLDs rather than learning a vendor's proprietary software package. Thus, Hardware Description Languages (HDLs) have emerged as the standard methodology for integrated-circuit and system design.

While the design community debated whether VHDL could become the standard for PLDs, Cypress took an industry leading position by introducing the first native VHDL compiler for programmable logic—our *Warp*™ software tools. Now, Cypress offers Verilog support.

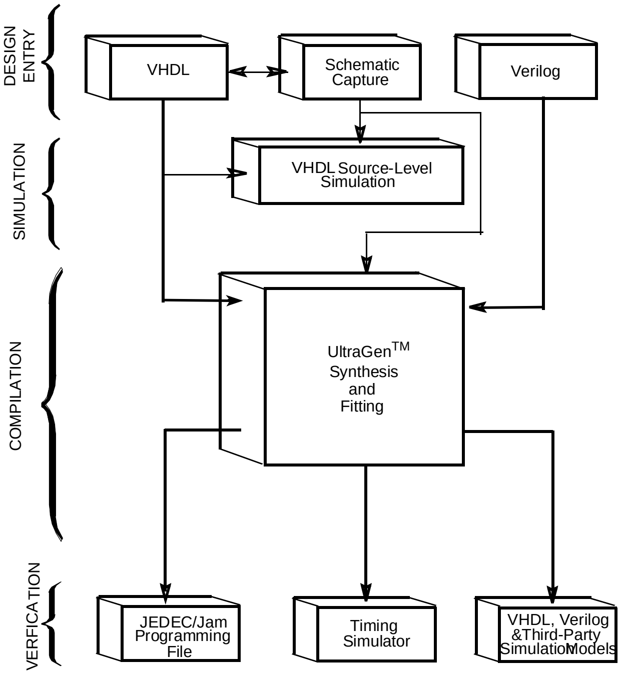

DESIGN ENTRY
VHDL
Schematic Capture
Verilog
SIMULATION
VHDL Source-Level Simulation
COMPILATION
UltraGen™ Synthesis and Fitting
VERFICATION
JEDEC/Jam Programming File
Timing Simulator
VHDL, Verilog &Third-Party SimulationModels

Figure 1. *Warp3* Design Flow.

Functional Description

Warp3 is an integration of Cypress's advanced VHDL and Verilog synthesis and fitting technology with Viewlogic's sophisticated EDA design environment. *Warp3* includes Cypress's VHDL and Verilog compiler and Workview Office software for Windows 95™, Windows 98™, and Windows NT™.

Design Flow

Figure 1 displays a block diagram of the typical design flow in *Warp3*. Designs can be entered in VHDL text, Verilog text, or schematic capture. *Warp3* also supports mixing of VHDL and schematics approaches on individual designs. VHDL designs are then functionally verified using the *Warp3* source-level simulator. The third step is to compile the design and target a PLD or CPLD. After synthesis, the waveform timing simulator is used to verify design timing as programmed in the chosen device. If the simulation results are satisfactory, the JEDEC is used to program the targeted device. A detailed description of each step follows.

Specifically, the *Warp3* Design Flow includes the following:

- Viewlogic GUI (Project Navigator)
- IEEE1076 and 1164 VHDL Synthesis
- IEEE1364 Verilog Synthesis
- Schematic Capture (ViewDraw)
- VHDL Source-level Simulator (SpeedWave)

July 28, 1999

- Hierarchy Navigator
- *Warp* Galaxy GUI for input
- Mixed-mode Design Entry for VHDL and Schematics
- Waveform Editor (ViewTrace)
- Timing Simulator (ViewSim or Active-HDL Sim)
- Finite State Machine Editor (Active-HDL FSM)
- Device fitters for all Cypress PLDs and CPLDs

Toolbars for WorkView Office

The Viewlogic Workview Office Toolbar is the Graphical User Interface (GUI) for accessing various tools. There are two toolbars, one for Workview Office, and one for *Warp*. *Figure 2* shows these two toolbars.

Design Entry

Text Editor

Through the *Warp* Galaxy Text Editor, text entry is done in either industry-standard VHDL or Verilog. *Warp3* can synthesize a rich set of the VHDL and Verilog languages in conformance with IEEE standard 1076, 1164, (VHDL) and 1364 (Verilog). This includes support for Behavioral, Boolean, State Table, and Structural VHDL and Verilog entry.

Text entry is ideal for describing complex logic functions such as state machines or truth tables. With VHDL or Verilog, the behavior of a state machine can be described in concise, easily-readable code. In addition, the hierarchical nature of VHDL and Verilog allow very complex functions to be described in a modular, top-down or bottom-up fashion. For more information on VHDL or Verilog see the *Warp2*® (CY3120 for VHDL or CY3110 for Verilog) data sheets.

Schematic Capture

Warp3 users can also enter designs graphically with a sophisticated schematic capture system (ViewDraw). With schematic entry, designers can quickly describe a variety of common logic functions from simple logic gates to complex arithmetic functions. See *Figure 3*.

Warp3 also supports the use of the LPM (Library of Parameterized Modules) standard for schematic library.

Within *Warp3*, users have access to an extensive symbol library of standard components and macro functions. These include:

- Adders/Subtractors
- Counters
- Comparators
- Multipliers
- Parity
- Shift Registers

In addition, the designer may create custom functions that can be used in any *Warp3* design.

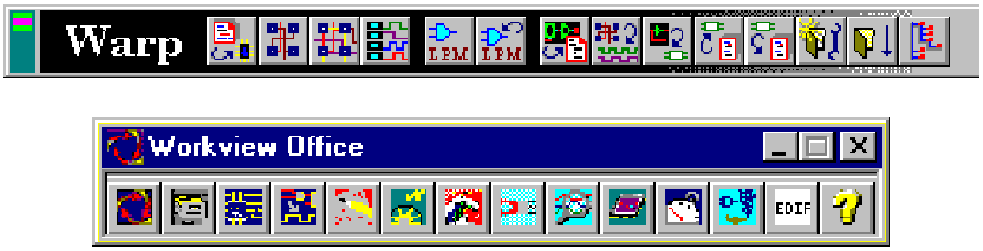
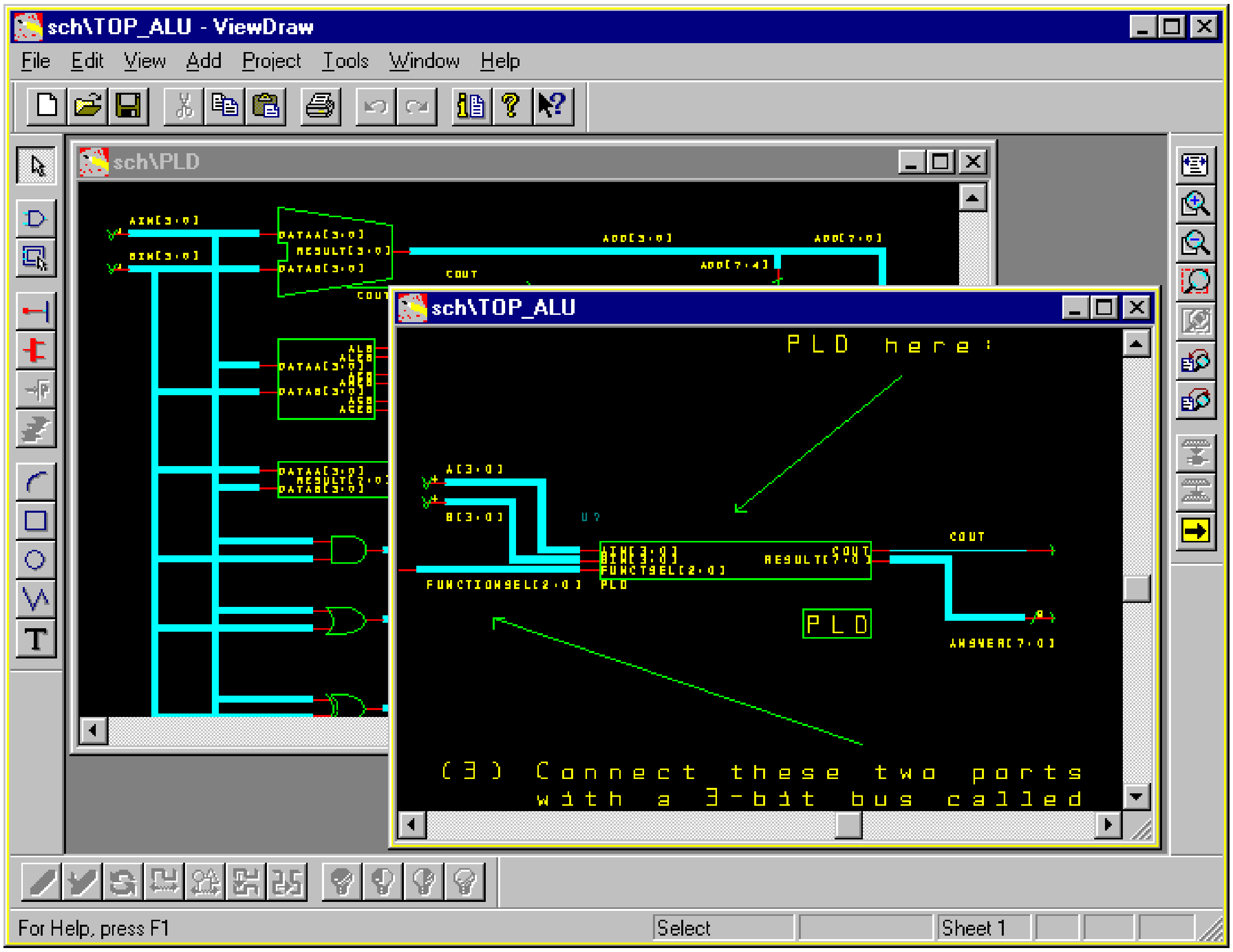


Figure 2. *Warp* and WorkView Office Toolbars for PC Platforms.



PLD here:

(3) Connect these two ports with a 3-bit bus called

Figure 3. Schematics with ViewDraw.

Symbol Editor

The *Warp3* schematic capture tools also provide methods to create symbols for schematics and VHDL blocks. Using the Viewgen utility, symbols are automatically generated from lower-level schematic data. Using the VHDL2SYM utility, symbols are automatically generated from VHDL text files. Symbols are useful for creating a design hierarchy to easily describe complex designs.

Mixed-Mode Entry

Perhaps the most powerful design entry methodology in *Warp3* is the combination of the above methods. In most designs, some portions of the circuit are most easily described in schematic form while others are best described in text. Typically, standard logic components such as counters, adders and registers are best implemented by retrieving components from the *Warp3* schematic symbol library. Meanwhile, text entry is usually preferred for describing sections of the circuit design that implement control logic. In particular, state machines are often much easier to describe with behavioral VHDL than with schematic components. Combining these methods in a single design simplifies the input process and shortens the design cycle time.

As mentioned before, *Warp3* can automatically generate symbols for text and schematic designs. This capability facilitates hierarchical design entry by allowing users to represent complex functions with a symbol. The top level of the design may be represented by the connection of a small number of symbols representing the main functional blocks. To move to lower levels in the design the user can push into selected symbols. If the underlying design is described in VHDL, a text window will be launched with the design file. If the underlying design is a schematic, a ViewDraw window will be opened with the schematic design. There is no limit to the number of levels of hierarchy used or the number of symbols in a particular design.

Finite State Machine Editor

The Finite State Machine Editor, Aldec-HDL FSM Editor, allows graphic design entry through the use of graphical state diagrams. The graphical state diagrams in conjunction with data flow logic represent HDL code.

Design Verification

Source-Level Simulation

Verifying functionality early in the design process can greatly reduce the number of design iterations necessary to complete a particular design. Using SpeedWave, the functionality of the design can be verified with textual stimulus from the keyboard or from a file. ViewTrace can be used in conjunction with SpeedWave to simulate the design functionality. The simulation process is described in detail below.

VHDL Source-Level Simulation

Speedwave is the powerful source-level VHDL simulator provided in *Warp3*. The VHDL debugger works in concert with the *Warp3* simulator and waveform editor. The simulator allows users to graphically step through VHDL code and monitor the results textually or in waveforms. After each single step the simulator highlights the VHDL text representing the current state of the simulation. Waveform and text windows can simultaneously display the inputs and outputs of the design.

Note that a design does not have to be entered in VHDL text to use the VHDL simulator. Since *Warp3* converts design schematics, etc. to VHDL before compilation, this VHDL representation can be single stepped to verify design functionality.

Hierarchy Navigator

Another powerful debugging tool within *Warp3* is the hierarchy navigator (Viewnav). The navigator allows users to select a net or node at one level of the design and automatically trace that net through all levels of the hierarchy. This is very useful for tracing signal paths when looking for design errors.

Compilation

VHDL Synthesis

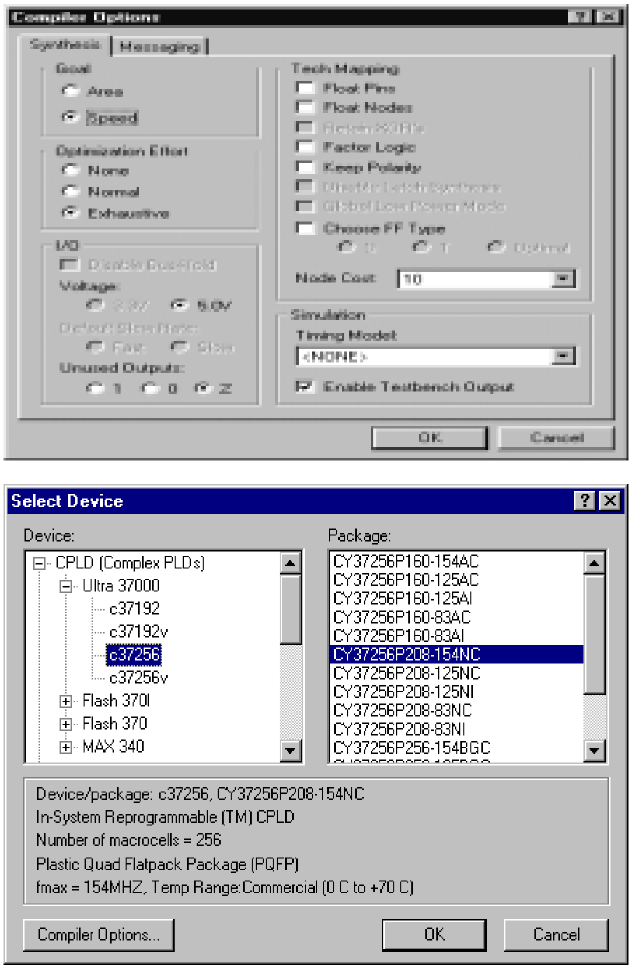
- ***Warp3* supports a rich subset of VHDL including:**
  - **— Enumerated types**
  - **— Integers**
  - **— For . . . Generate loops**
  - **— Operator overloading**

Verilog Synthesis

- ***Warp3* supports a rich subset of Verilog including:**
  - **— Reduction and Conditional operators**
  - **— Blocking and Non-Blocking procedural assignments**
  - **— While loop**
  - **— Integers**

Once design entry is complete and functionality has been verified, the entire design is converted to VHDL using the “Export 1164” utility on schematic modules. At this point in the design there is a VHDL description of the entire design. This VHDL description is fed to the Cypress VHDL compiler for logic optimization, fitting, and translation to a device programming file. The same is done for Verilog descriptions. Although compilation is a multi-step process, it appears as a single step to the user (as shown in *Figure 1*).

The first step in compilation is synthesizing the input VHDL or Verilog into a logical representation of the design in terms of components found in the target device (AND gates, OR gates, flip-flops etc.). *Warp3* synthesis is unique in that the input languages (VHDL or Verilog) support device-independent design descriptions.



Compiler Options

Synthesis | Messaging

Goal: ○ Area ● Speed

Optimization Effort: ○ None ○ Normal ● Exhaustive

I/O: □ Disable Bus-Hold

Voltage: ○ 3.3V ● 5.0V

Default Slew Rate: ○ Fast ○ Slow

Unused Outputs: ○ 1 ○ 0 ● Z

Tech Mapping: □ Float Pins □ Float Nodes □ Retain XOR's □ Factor Logic □ Keep Polarity □ Disable Latch Synthesis □ Global Low Power Mode □ Choose FF Type ○ D ○ T ○ Optimal

Node Cost: 10

Simulation: Timing Model: <NONE>

☑ Enable Testbench Output

OK | Cancel

Select Device

Device:

- CPLD (Complex PLDs)
  - Ultra 37000
    - c37192
    - c37192v
    - c37256
    - c37256v
  - Flash 370I
  - Flash 370
  - MAX 340

Package:

CY37256P160-154AC
CY37256P160-125AC
CY37256P160-125AI
CY37256P160-83AC
CY37256P160-83AI
CY37256P208-154NC
CY37256P208-125NC
CY37256P208-125NI
CY37256P208-83NC
CY37256P208-83NI
CY37256P256-154BGC

Device/package: c37256, CY37256P208-154NC
In-System Reprogrammable (TM) CPLD
Number of macrocells = 256
Plastic Quad Flatpack Package (PQFP)
fmax = 154MHZ, Temp Range:Commercial (0 C to +70 C)

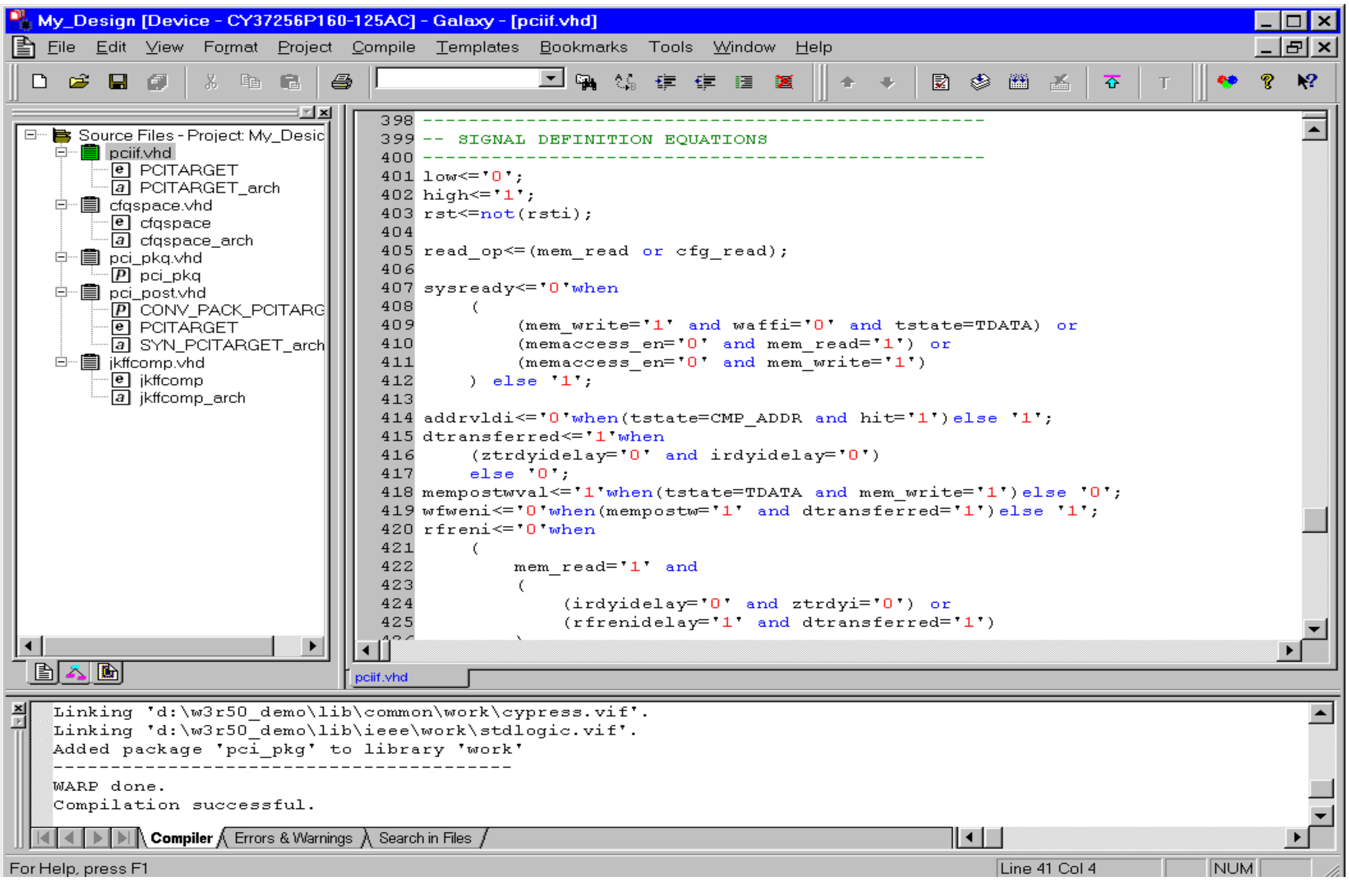
Compiler Options... | OK | Cancel

Figure 4. Compiler Options.

- **State-of-the-art optimization and reduction algorithms including:**
  - **— Optimization for flip-flop type (D type/T type)**
  - **— Automatic pin assignment**
  - **— User-specified state assignment (Gray code, binary, one-hot)**

For PLDs and CPLDs, the second phase of the compilation is an iterative process of optimizing the design and fitting the logic into the targeted device as shown in *Figure 1*. Logical optimization in *Warp3* is accomplished with Espresso algorithms. Once optimized, the design is fed to the device-specific fitter which applies the design to the selected device, as shown in *Figure 4*. *Warp3* fitters support manual or automatic pin assignments as well as automatic selection of D-type or T-type flip-flops. After optimization and fitting are complete, *Warp3* will create a JEDEC file used to program the device.

To facilitate design conversion, a JEDEC translator is provided to convert FLASH370i JEDEC files to JEDEC files that target Ultra37000 devices.



```
------------------------------------------------
-- SIGNAL DEFINITION EQUATIONS
------------------------------------------------
low<='0';
high<='1';
rst<=not(rsti);

read_op<=(mem_read or cfg_read);

sysready<='0'when
    (
        (mem_write='1' and waffi='0' and tstate=TDATA) or
        (memaccess_en='0' and mem_read='1') or
        (memaccess_en='0' and mem_write='1')
    ) else '1';

addrvldi<='0'when(tstate=CMP_ADDR and hit='1')else '1';
dtransferred<='1'when
    (ztrdyidelay='0' and irdyidelay='0')
    else '0';
mempostwval<='1'when(tstate=TDATA and mem_write='1')else '0';
wfweni<='0'when(mempostw='1' and dtransferred='1')else '1';
rfreni<='0'when
    (
        mem_read='1' and
        (
            (irdyidelay='0' and ztrdyi='0') or
            (rfrenidelay='1' and dtransferred='1')
```

```
Linking 'd:\w3r50_demo\lib\common\work\cypress.vif'.
Linking 'd:\w3r50_demo\lib\ieee\work\stdlogic.vif'.
Added package 'pci_pkg' to library 'work'
---------------------------------------
WARP done.
Compilation successful.
```

Figure 5. Compile Dialog Box and Color-Coded Editor.

Automatic Error Tracking

Of course, the compilation process may not always go as planned. VHDL and/or Verilog syntax errors should be identified and corrected in the pre-synthesis functional simulation stage. During the compilation phase, *Warp3* will detect errors that occur in the fitting process. *Warp3* features automatic error location that allows problems to be diagnosed and corrected in seconds. Errors from compilation are displayed immediately in a window (see *Figure 5*). If the user highlights a particular error, *Warp3* will automatically open the source code file and highlight the offending line in the entered design. If the device fitting process includes errors, a window will again describe them. In addition, a detailed report file is generated indicating the resources required to fit the input design and any problems that occurred in the process.

Simulation

The last step in the design process before programming is verifying the timing of your design. For this, *Warp3* includes the ViewSim timing simulator. During compilation, delays that result from fitting the input design are “written” into an internal file for use by the *Warp3* simulator. This information represents worst-case path delays for the design as fit in the selected device. Delays are based on the type of device and speed grade selected.

One of the ways to simulate is with the command-line interface to ViewSim. From the command line, the designer can specify the state of inputs (high, low, X, etc.) and watch how outputs behave over a specified time frame. In this way users can easily step through test cases and view the output results. Stimulus can be entered from the command line or from a file.

In addition to ViewSim and SpeedWave, *Warp3* will output VHDL and Verilog timing models for numerous other simulators:

- VHDL
  - — Model Technology/Mentor Graphics (V-System/ QuickHDL)
- Verilog
  - — Cadence (Verilog XL)
  - — Model Technology/Mentor Graphics (V-System)
  - — Veribest (Interlog)
  - — Viewlogic (VCS)

Waveform Editor

A graphical method of simulation uses the Viewlogic waveform editor, ViewTrace, in conjunction with ViewSim or SpeedWave. With ViewTrace users can input stimulus from a file or graphically via digital waveforms. Outputs are viewed as digital waveforms that reflect the timing delays of the device as programmed. ViewTrace is interactive, allowing modifications of the stimulus and re-simulation of the results without re-running synthesis tools.

If user inputs violate device specifications the *Warp3* simulator will detect the violation and warn the user. For example, if an input changes immediately before a CLK rise (violating the device set-up time) *Warp3* will issue a warning and highlight the offending signal. The same occurs for all other timing violations.

Active-HDL Sim

Active-HDL Sim is a post-synthesis timing simulator that features a graphical waveform simulator that can be used to simulate PLD/CPLD designs generated in *Warp3.* The simulator provides timing simulation for PLDs/CPLDs and features interactive waveform editing and viewing. The simulator also provides the ability to probe internal nodes, and automatically generate clocks and pulses.

Programming

The result of *Warp* compilation is a JEDEC file that implements the input design in the targeted device. Using this file, Cypress devices can be programmed on any qualified third-party programmer.

Cypress's Ultra37000 and FLASH370i In-System Reprogrammable™ (ISR™) devices can also be programmed on board with an ISR programmer. Using Cypress's ISR Programming Software, Flash370i devices can be programmed with the JEDEC file. Ultra37000 files can be programmed via Jam files. Cypress's ISR software converts JEDEC files into Jam files which can then be used to program the device. For more information on the use of the ISR software, refer to the data sheet for either the Ultra37000 ISR Programming Kit (CY3700i) or the FLASH370i ISR Programming Kit (CY3600i).

System Requirements

PC Platform

- IBM PC or equivalent (Pentium® class recommended)
- 32 Mbytes of RAM (64 Mbytes recommended)
- 110 Mbytes Disk Space
- CD-ROM drive
- Windows 95, Windows 98, or Windows NT 4.0

Ordering Information

CY3130R52 *Warp3* PLD Development System on the PC includes:

- CD-ROM with *Warp3* software and on-line documentation
- CD-ROM with Workview Office software and on-line documentation
- *Warp3* Viewlogic hardware key
- Getting Started Manual
- User's Guide
- HDL Reference Guide
- Release Notes
- *VHDL for Programmable Logic* textbook
- License and registration form

Document #: 38-00242-K

Ultra37000*,* FLASH370i*, Warp,* In-System Reprogrammable, and ISR are trademarks of Cypress Semiconductor Corporation.
Warp2 and *Warp3* are registered trademarks of Cypress Semiconductor Corporation.
Pentium is a registered trademark of Intel Corporation.
Windows 95, Windows 98, and Windows NT are trademarks and Windows is a registered trademark of Microsoft Corporation.
Workview Office, ViewDraw, ViewSim, ViewTrace, ViewNav, and SpeedWave are registered trademark of Viewlogic Systems, Inc.
Active-HDL is a trademark of Aldec Incorporated.