Warp3

PRELIMINARY **(CY3130/CY3135)**

Warp3™ VHDL Development System for PLDs, CPLDs, and FPGAs

Features

- **Sophisticated PLD/FPGA design and verification system based on VHDL**
- ***Warp3*™ is based on Viewlogic's Powerview™ (Sun and HP) and Workview Plus™ (PC) design environments**
  - **— Advanced graphical user interface for Windows and Sun/HP Workstations**
  - **— Schematic capture (Viewdraw)**
  - **— Interactive timing simulator (Viewsim)**
  - **— Waveform stimulus and viewing (Viewtrace)**
  - **— Textual design entry using VHDL**
  - **— Mixed-mode design entry support**
- **The core of *Warp3* is an IEEE1076 and 1164 standard VHDL compiler**
  - **— VHDL is an open, powerful design language**
  - **— VHDL (IEEE standard 1076 and 1164) facilitates design portability across devices and/or CAD platforms**
  - **— VHDL facilitates the use of industry-standard simulation and synthesis tools for board and system-level design**
  - **— VHDL facilitates hierarchical design with support for functions and libraries**
- **Support for ALL Cypress PLDs, CPLDs, FPGAs, and PROMs, including:**
  - **— Industry-standard 20- and 24-pin devices like the 22V10**
  - **— Cypress 7C33X family of 28-pin PLDs**
  - **— MAX340 (MAX5000 Series) CPLDs**
  - **— FLASH370™ CPLDs**
  - **— pASIC380™ FPGAs**

Introduction

As the capacity and complexity of programmable logic increased dramatically over the last couple of 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) in general, and VHDL (Very high speed integrated-circuit Hardware Description Language) in particular, 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.

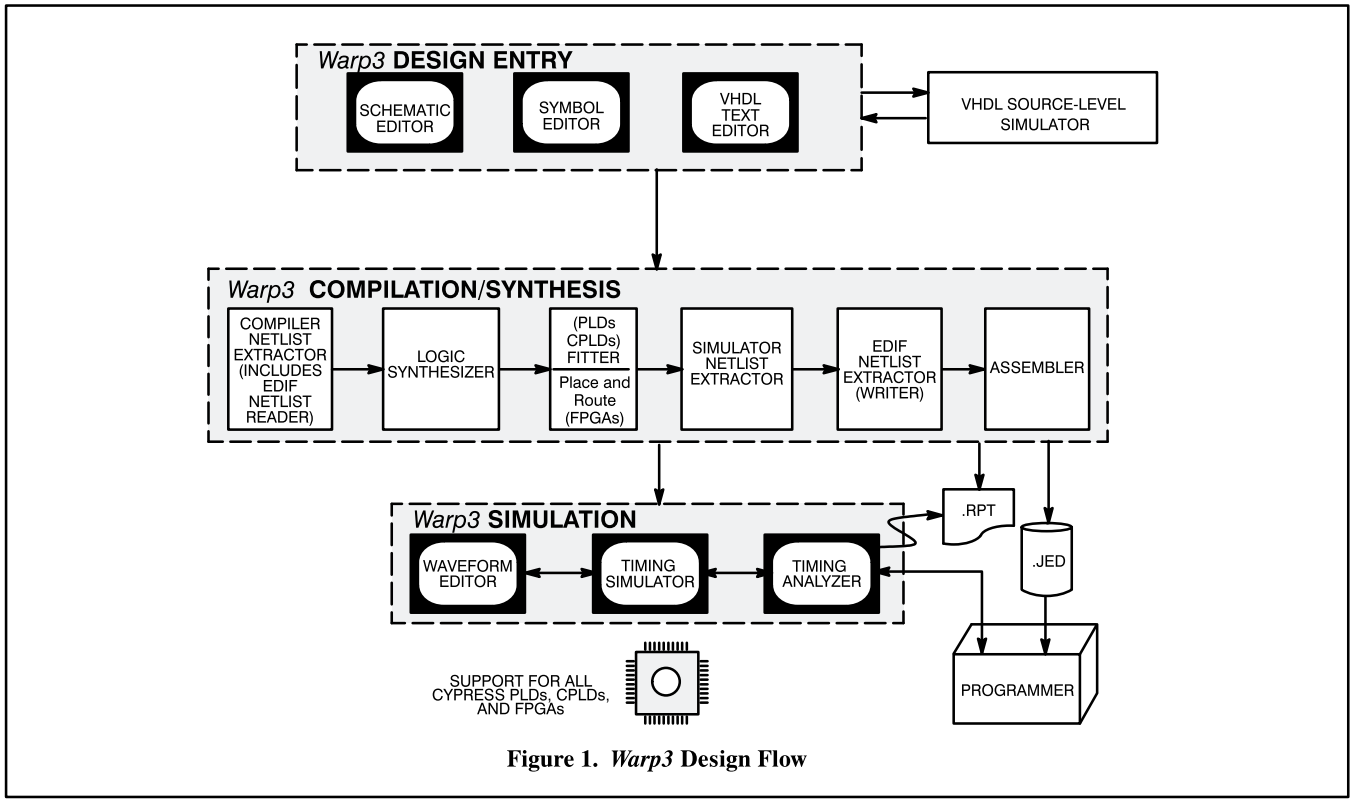

Figure 1. *Warp3* Design Flow

Functional Description

Warp3 is an integration of Cypress's advanced VHDL synthesis and fitting technology with Viewlogic's sophisticated CAE design environment. On the PC platform, *Warp3* includes Cypress' VHDL compiler and Viewlogic's Workview Plus software for Microsoft Windows. On the Sun and HP platforms, *Warp3* includes Cypress' VHDL compiler and Viewlogic's Powerview software.

Design Flow

Figure 1 displays a block diagram of the typical design flow in *Warp3*. Designs can be entered in VHDL text, schematic capture or via an imported EDIF netlist. In fact, *Warp3* supports mixing these approaches on individual designs. Designs are then functionally verified using the *Warp3* functional simulator. The third step is to compile the design and target a PLD, CPLD or FPGA. 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 or netlist file is used to program the targeted device. A detailed description of each step follows.

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

Viewlogic GUI (Cockpit)
IEEE1076 and 1164 VHDL Synthesis
Schematic Capture (Viewdraw)
Hierarchy Navigator
Mixed-mode Design Entry
Waveform Editor (Viewtrace)
VHDL Timing Simulator (Viewsim)
Device fitters for all Cypress PLD/CPLDs/PROMs
Automatic Place&Route for all Cypress FPGAs

The Cockpit

The Viewlogic graphical user interface (GUI) is built around a file/tool manager called "the cockpit". The cockpit is used to select the project and current tool set in use. The cockpit allows users to select from a variety of design environments called toolboxes. For UNIX workstations the GUI is under the PowerView cockpit and for PC/Windows the GUI is under the WorkView Plus cockpit (see *Figure 2*).

Design Entry

Text Editor

Text entry is done with industry standard VHDL. *Warp3* can synthesize a rich set of the VHDL language in conformance with IEEE standard 1076 and 1164. This includes support for Behavioral, Boolean, State Table and Structural VHDL entry.

Text entry is ideal for describing complex logic functions such as state machines or truth tables. With VHDL, the behavior of a state machine can be described in concise, easily-readable code. In addition, the hierarchical nature of VHDL allows very complex functions to be described in a modular, top-down or bottom-up fashion. For more information on VHDL see the *Warp2*™ (CY3121) or the *Warp2+*™ (CY3120) datasheet.

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.

Warp3 also supports the use of the LPM (Library of Parameterized Modules) Standard. With LPM, users can import schematic and text designs from any CAE tool that supports the LPM standard via an EDIF netlist. Designs based on this standard library will then be portable across many CAE and synthesis (see *Figure 3*) platforms.

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

- adders/multipliers
- counters
- gates (AND, OR, XOR, INV, & BUF)
- io (singles, buses, three-states, clk-pads, hd-pad, gnd, & vcc)
- macrocells
- memory (assorted flip-flops and latches)
- mux (decoders and multiplexers)
- registers, shift registers and universal registers

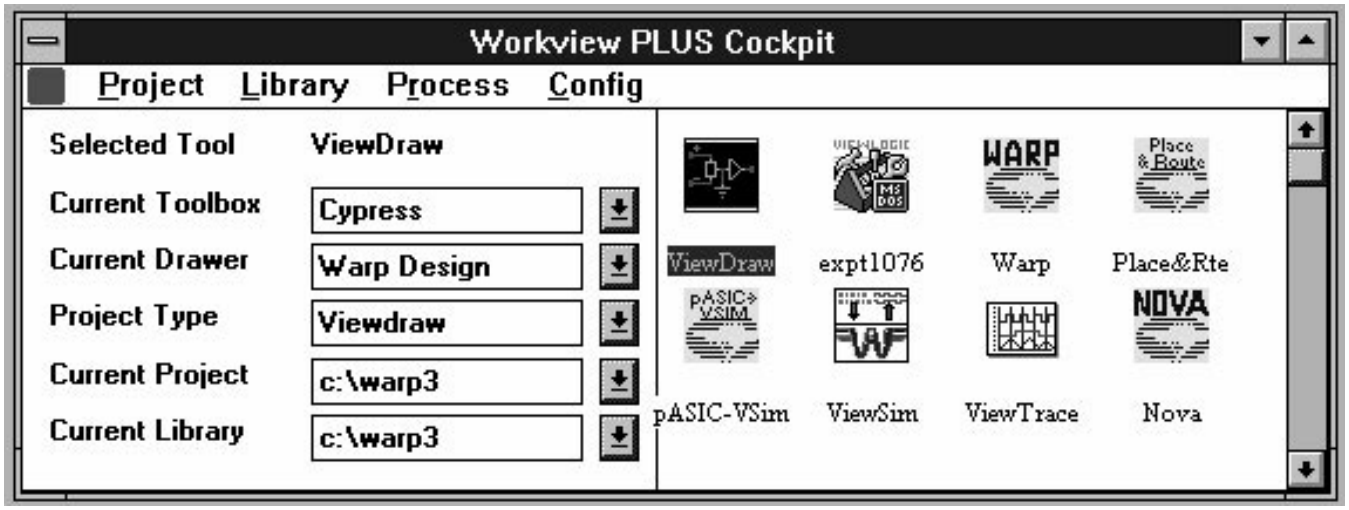

Figure 2. WorkView PLUS Cockpit for PC Workstations

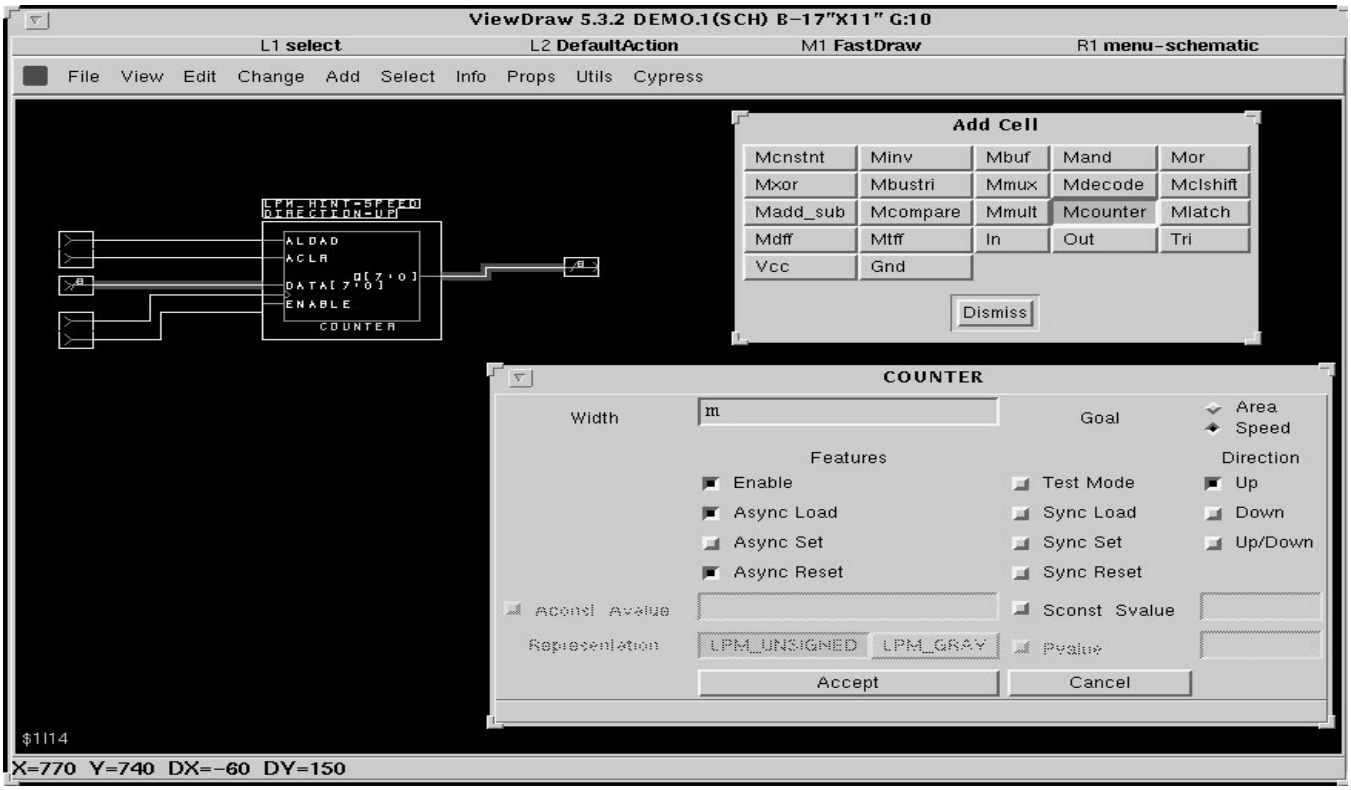

Figure 3. Using the LPM Symbol Library in Viewdraw

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

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.

EDIF Input

Warp3 includes an EDIF netlist converter that provides a convenient way for designers to import designs from other CAE schematic capture and simulation tools. The EDIF-in tool supports EDIF version 2 0 0.

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 above, *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.

Design Verification

Functional 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 graphically. The simulation process is described in detail below.

VHDL Source-level Simulation

A unique and powerful feature of *Warp3* is the source-level VHDL simulator. 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. Simultaneously waveform and text windows can 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 all facets of a design (schematic, EDIF-in 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

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

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. Although compilation is a multistep 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 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 language (VHDL) supports device-independent design descriptions. Competing programmable logic compilers require very specific and device-dependent information in the design file.

Device Fitting

- **State-of-the-art optimization and reduction algorithms**
  - **— 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 (see *Figure 4*). 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 (see *Figure 5*). *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 (PLDs and CPLDs) used to program the device.

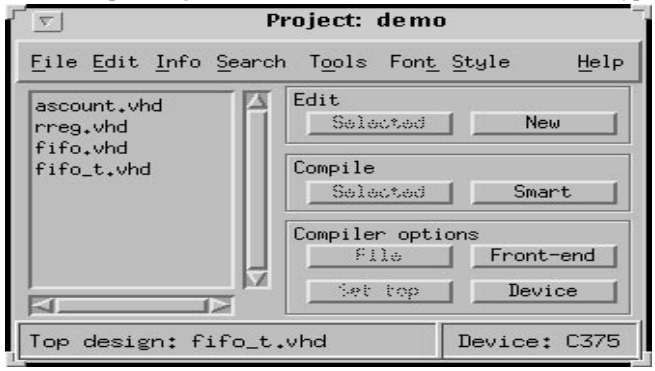

Figure 4. Compile/Synthesize Dialog Box

Automatic Place&Route

- **Completely automatic place and route**
  - **— Includes timing back annotation into Viewsim**

For Cypress FPGAs, the second phase of the compilation process is called place&route. The place&route tools in *Warp3* take the logical design description from synthesis and apply it to the cells of the targeted FPGA. Once placed, the programmable interconnect channels are routed to connect logic blocks as required by the design. With Cypress FPGAs and *Warp3*, the place&route process is 100% automatic. No tedious manual intervention or hand tweaking is necessary. Once place&route is finished, *Warp3* generates a netlist that is used to program the FPGA or a device programmer.

Automatic Error Tracking

Of course, the compilation process may not always go as planned. VHDL 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/place&route process. *Warp3* features automatic error location that allows problems to be diagnosed and corrected in seconds. Errors from compilation are displayed immediately in a pop-up window. 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 or place&route process includes errors, a pop-up 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 VHDL 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

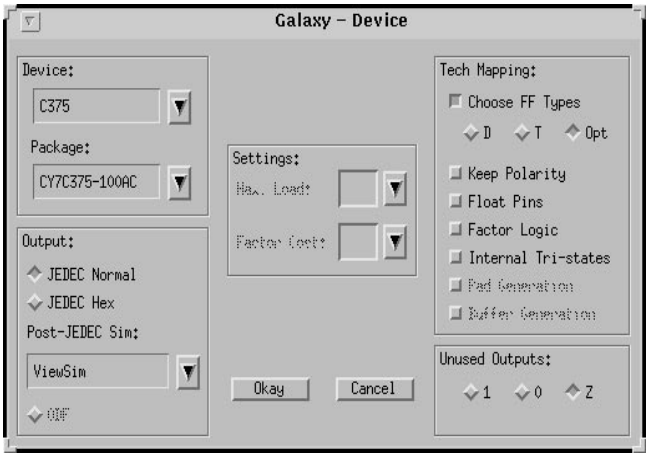

Figure 5. Device Fitting/Routing Dialog Box

through test cases and view the output results. Stimulus can be entered from the command line or from a file.

In addition to Viewsim, *Warp3* will output VHDL and Verilog timing models for numerous other simulators.

- VHDL
  - — Mentor (Model Technology and Quicksim)
  - — Cadence (Leapfrog)
  - — Synopsys (V_{SS})
  - — Viewlogic (Vantage)
- Verilog
  - — Cadence
  - — Intergraph

Waveform Editor

A graphical method of simulation uses the Viewlogic waveform editor, Viewtrace, in conjunction with Viewsim. 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.

Programming

After the design is compiled and verified, the targeted device is ready for programming. The program file generated in *Warp3* (a JEDEC file or LOF file) is used as input to a device programmer. Cypress offers an inexpensive programmer, the Impulse3™ based on Data I/O's ChipLab™, that programs all Cypress PLDs and FPGAs. Alternatively, customers can use any one of several qualified 3rd party programmers from corporations like Data I/O, SMS and Logical Devices.

System Requirements

PC Platform

80486-based IBM PC
Microsoft Windows V3.1
16 Mbytes of RAM
110 Mbytes Disk Space
1.44-Mbyte 3.5-inch floppy disk drive

Sun Platform

SPARC CPU
Sun OS 4.1.1 or later
Motif GUI
16 Mbytes of RAM
120 Mbytes of Disk Space
Cartridge Tape

HP Platform

HP700 Series CPU
OS 9.05
Motif GUI
16 Mbytes of RAM
130 Mbytes of Disk Space
Cartridge Tape

Ordering Information

CY3130 *Warp3* PLD Development System on the PC includes:
- 3 1/2-inch 1.44-Mbyte floppy disks
- *Warp3* Viewlogic hardware key
- *Warp3* User's Guide
- *Warp3* Reference Manual
- Registration Card

CY3131[2] *Warp3* PLD Development System on the PC (for current Viewlogic Users of Workview Plus) includes:
- 3 1/2-inch 1.44-Mbyte floppy disks
- *Warp3* User's Guide
- *Warp3* Reference Manual
- Registration Card

CY3135 *Warp3* PLD Development System on a UNIX/SUN Workstation includes:
- Three Cartridge Tapes
  - 1) Viewlogic Software
  - 2) *Warp3* Software
  - 3) Viewlogic On-line Documentation
- *Warp3* User's Guide
- *Warp3* Reference Manual
- Registration Card

CY3136[3] *Warp3* PLD Development System on a UNIX/SUN Workstation (for current Viewlogic Users of Powerview) includes:
- One Cartridge Tapes of *Warp3* Software
- *Warp3* User's Guide
- *Warp3* Reference Manual
- Registration Card

Notes:

2. This is a "Bolt-in" Solution and requires the customer to be a current User of Viewlogic's Workview Plus S/W
3. This is a "Bolt-in" Solution and requires the customer to be a current User of Viewlogic's Powerview S/W.

Document #: 38−00242−D