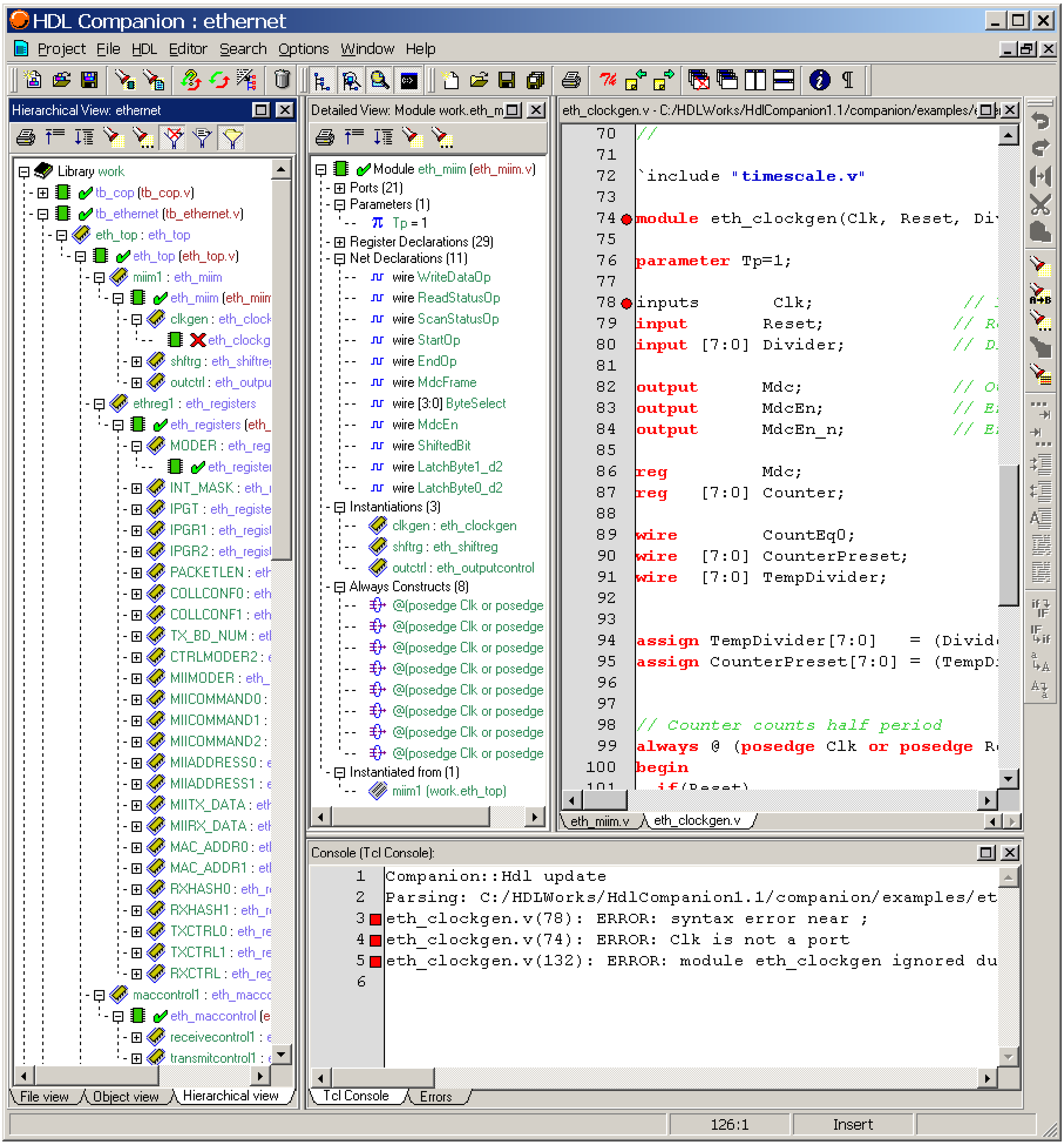
HDL Companion


HDL Companion : ethernet
Project File HDL Editor Search Options Window Help
Hierarchical View: ethernet
Detailed View: Module work.eth_m
eth_clockgen.v - C:/HDLWorks/HdlCompanion1.1/companion/examples/
Console (Tcl Console):
File view
Object view
Hierarchical view
Tcl Console
Errors
eth_miim.v
eth_clockgen.v
126:1
Insert

```
Library work
  tb_cop (tb_cop.v)
  tb_ethernet (tb_ethernet.v)
    eth_top : eth_top
      eth_top (eth_top.v)
        miim1 : eth_miim
          eth_miim (eth_miim
            clkgen : eth_clock
              eth_clockg
            shftrg : eth_shiftre
            outctrl : eth_outpu
        ethreg1 : eth_registers
          eth_registers (eth_
            MODER : eth_reg
              eth_registe
            INT_MASK : eth_
            IPGT : eth_registe
            IPGR1 : eth_regis
            IPGR2 : eth_regis
            PACKETLEN : eth
            COLLCONF0 : eth
            COLLCONF1 : eth
            TX_BD_NUM : et
            CTRLMODER2 : e
            MIIMODER : eth_
            MIICOMMAND0 :
            MIICOMMAND1 :
            MIICOMMAND2 :
            MIIADDRESS0 : e
            MIIADDRESS1 : e
            MIITX_DATA : eth
            MIIRX_DATA : eth
            MAC_ADDR0 : et
            MAC_ADDR1 : et
            RXHASH0 : eth_r
            RXHASH1 : eth_r
            TXCTRL0 : eth_re
            TXCTRL1 : eth_re
            RXCTRL : eth_reg
        maccontrol1 : eth_macco
          eth_maccontrol (e
            receivecontrol1 : e
            transmitcontrol1 : e
```

```
Module eth_miim (eth_miim.v)
  Ports (21)
  Parameters (1)
    Tp = 1
  Register Declarations (29)
  Net Declarations (11)
    wire WriteDataOp
    wire ReadStatusOp
    wire ScanStatusOp
    wire StartOp
    wire EndOp
    wire MdcFrame
    wire [3:0] ByteSelect
    wire MdcEn
    wire ShiftedBit
    wire LatchByte1_d2
    wire LatchByte0_d2
  Instantiations (3)
    clkgen : eth_clockgen
    shftrg : eth_shiftreg
    outctrl : eth_outputcontrol
  Always Constructs (8)
    @(posedge Clk or posedge
    @(posedge Clk or posedge
    @(posedge Clk or posedge
    @(posedge Clk or posedge
    @(posedge Clk or posedge
    @(posedge Clk or posedge
    @(posedge Clk or posedge
    @(posedge Clk or posedge
  Instantiated from (1)
    miim1 (work.eth_top)
```

```
//

`include "timescale.v"

module eth_clockgen(Clk, Reset, Di

parameter Tp=1;

inputs       Clk;              // 
input       Reset;             // R
input  [7:0] Divider;          // D

output      Mdc;               // O
output      MdcEn;             // E
output      MdcEn_n;           // E

reg         Mdc;
reg    [7:0] Counter;

wire        CountEq0;
wire   [7:0] CounterPreset;
wire   [7:0] TempDivider;


assign TempDivider[7:0]    = (Divid
assign CounterPreset[7:0] = (TempD


// Counter counts half period
always @ (posedge Clk or posedge R
begin
  if(Reset)
```

```
Companion::Hdl update
Parsing: C:/HDLWorks/HdlCompanion1.1/companion/examples/et
eth_clockgen.v(78): ERROR: syntax error near ;
eth_clockgen.v(74): ERROR: Clk is not a port
eth_clockgen.v(132): ERROR: module eth_clockgen ignored du
```

HDL Works

www.hdlworks.com

the ideal environment for

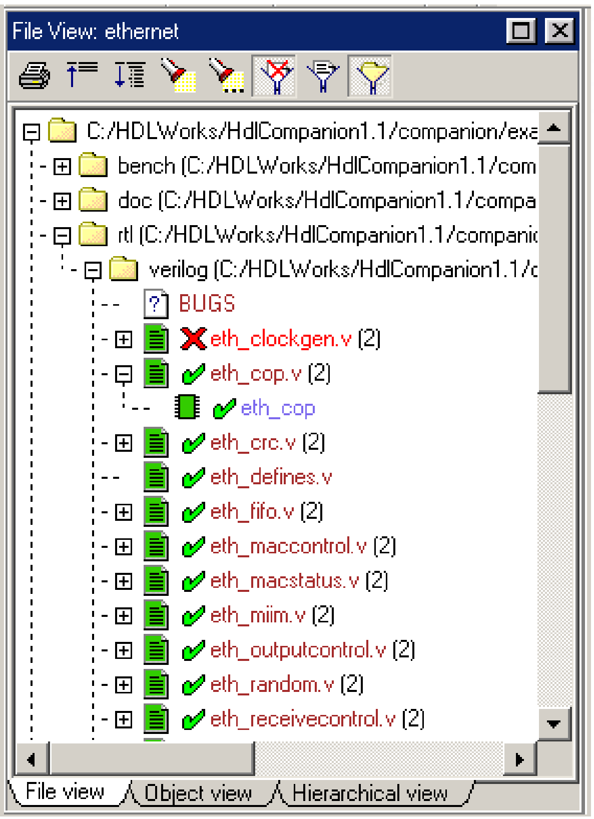
providing insight in IP
combining IP with new code
keeping overview of complex HDL designs

The complexity of digital designs has increased dramatically over the past decade. Today an FPGA or ASIC project is done by a team of engineers and consists of third party IP, internal re-use and newly developed code. HDL Companion has been developed to improve the productivity of the HDL design engineer.

In the past a designer had to use a simple text editor and Unix like ‘grep’ commands to find his way through many lines of HDL code or use a costly simulator license to understand the design structure. With HDL Companion a designer can drag and drop design files or directories into the file view and a complete design decomposition is performed in seconds, offering information regarding numerous aspects of the design.

HDL Companion

HDL Companion consists of four windows including a command console. Together they offer a complete overview of your design, from library level downto the details of your HDL source code. Each window can be enabled and disabled separately so you can easily configure the tool to show you exactly the information you are focusing on. HDL Companion is equiped with both fuzzy and fully compliant VHDL and Verilog parsers, freeing you from the burden of determining file dependencies while offering you full syntax checking.



```
⊟ C:/HDLWorks/HdlCompanion1.1/companion/exa
  ⊞ bench (C:/HDLWorks/HdlCompanion1.1/com
  ⊞ doc (C:/HDLWorks/HdlCompanion1.1/compa
  ⊟ rtl (C:/HDLWorks/HdlCompanion1.1/companic
    ⊟ verilog (C:/HDLWorks/HdlCompanion1.1/c
        BUGS
      ⊞ ✗ eth_clockgen.v (2)
      ⊟ ✓ eth_cop.v (2)
          ✓ eth_cop
      ⊞ ✓ eth_crc.v (2)
        ✓ eth_defines.v
      ⊞ ✓ eth_fifo.v (2)
      ⊞ ✓ eth_maccontrol.v (2)
      ⊞ ✓ eth_macstatus.v (2)
      ⊞ ✓ eth_miim.v (2)
      ⊞ ✓ eth_outputcontrol.v (2)
      ⊞ ✓ eth_random.v (2)
      ⊞ ✓ eth_receivecontrol.v (2)
```

Global Window

The Global Window is separated into three tab pages offering you a file view, an object view and a hierarchical design view.

The file view shows you the files present in your design in a browser like manner while also presenting the main design objects contained in each file. You can select any object to either edit it in the Source Window (or your preferred editor) or have a more detailed description in the Detailed Window. The file view is not limited to HDL files, but shows all files in your project, providing easy access to all your design data, like PDF and other documents. It is also the place to perform version management actions when a version management system is enabled. File view filters allow you to focus on what you want to see.

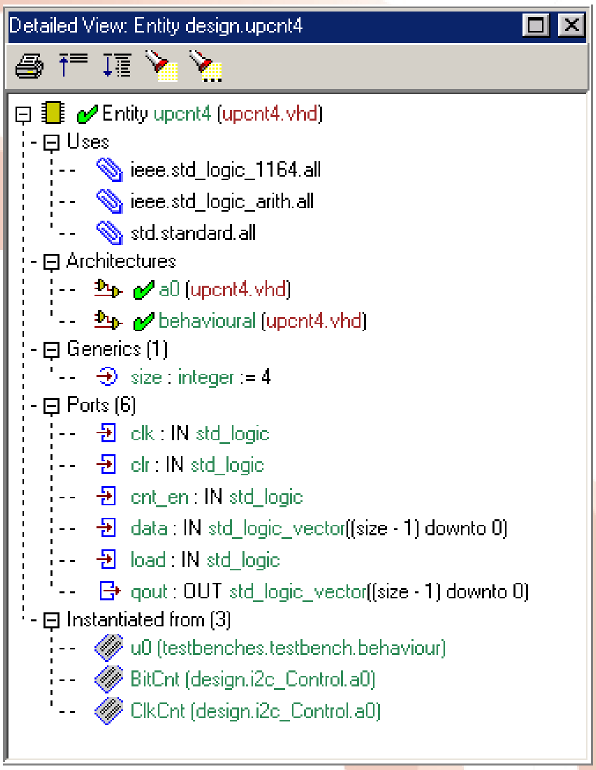
The object view shows all objects in the design (entities, architectures, configurations, modules, etc) organized by HDL library. The hierarchical view shows the hierarchy of the selected design.

The views are driven by fuzzy VHDL and Verilog parsers. These parsers take care of all the tedious work of placing objects in libraries and the file compilation order. They also allow you to work with incomplete or erroneous designs.

All views are related and allow drag and drop for easy navigation through your project.

Detailed Window

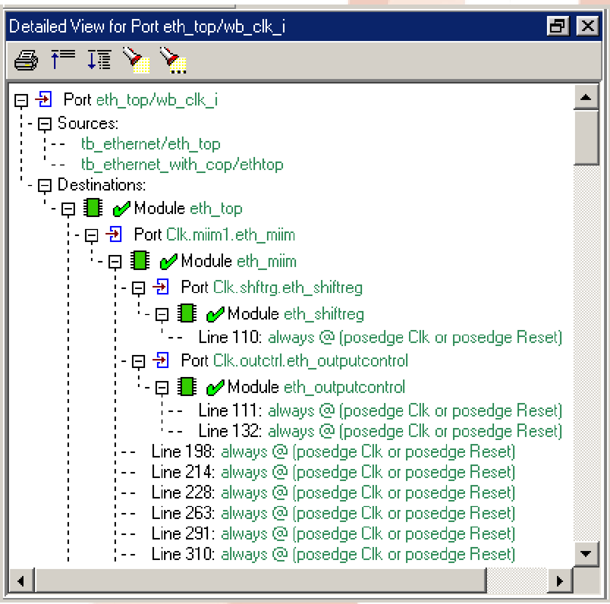
The Detailed Window offers you an in-depth view of the structure or hierarchy of a selected HDL object. For an entity or module the list of parameters, ports and architectures is shown, as well as the places where the entity or module is instantiated.



Detailed View: Entity design.upcnt4

```
⊟ ✓ Entity upcnt4 (upcnt4.vhd)
  - ⊟ Uses
      -- ieee.std_logic_1164.all
      -- ieee.std_logic_arith.all
      -- std.standard.all
  - ⊟ Architectures
      -- ✓ a0 (upcnt4.vhd)
      -- ✓ behavioural (upcnt4.vhd)
  - ⊟ Generics (1)
      -- size : integer := 4
  - ⊟ Ports (6)
      -- clk : IN std_logic
      -- clr : IN std_logic
      -- cnt_en : IN std_logic
      -- data : IN std_logic_vector((size - 1) downto 0)
      -- load : IN std_logic
      -- qout : OUT std_logic_vector((size - 1) downto 0)
  - ⊟ Instantiated from (3)
      -- u0 (testbenches.testbench.behaviour)
      -- BitCnt (design.i2c_Control.a0)
      -- ClkCnt (design.i2c_Control.a0)
```

The Detailed Window is also used to show the trace results of a port or signal.



Detailed View for Port eth_top/wb_clk_i

```
⊟ Port eth_top/wb_clk_i
  - ⊟ Sources:
      -- tb_ethernet/eth_top
      -- tb_ethernet_with_cop/ethtop
  - ⊟ Destinations:
      - ⊟ ✓ Module eth_top
          - ⊟ Port Clk.miim1.eth_miim
              - ⊟ ✓ Module eth_miim
                  - ⊟ Port Clk.shftrg.eth_shiftreg
                      - ⊟ ✓ Module eth_shiftreg
                          -- Line 110: always @ (posedge Clk or posedge Reset)
                  - ⊟ Port Clk.outctrl.eth_outputcontrol
                      - ⊟ ✓ Module eth_outputcontrol
                          -- Line 111: always @ (posedge Clk or posedge Reset)
                          -- Line 132: always @ (posedge Clk or posedge Reset)
                  -- Line 198: always @ (posedge Clk or posedge Reset)
                  -- Line 214: always @ (posedge Clk or posedge Reset)
                  -- Line 228: always @ (posedge Clk or posedge Reset)
                  -- Line 263: always @ (posedge Clk or posedge Reset)
                  -- Line 291: always @ (posedge Clk or posedge Reset)
                  -- Line 310: always @ (posedge Clk or posedge Reset)
```

This gives you detailed information about where and how signals are used throughout the whole hierarchy of your design. All labels in the treeview are clickable, providing easy access to the source code.

Source Window

The Source Window offers a fully featured language sensitive text editor with a multiple document interface. You can avoid typing errors and improve your productivity by using language templates, identifier repeat and one-touch line and column manipulation. Syntax highlighting keeps your text highly readable and well structured. The marker system allows you to drop markers for fast navigation between various file and text positions. Syntax errors or warnings are shown by colored dots at the beginning of a line. Tight integration with the data model allows you to directly jump to object definitions and their use.

If you want to use your own favorite text editor, you can configure HDL Companion to use the integrated editor for viewing only.

Console Window

The Console Window shows all errors, warnings and messages. Double clicking on error or warning messages enables hot links to the HDL source code. The Console Window also functions as a Tcl interpreter, allowing you to execute any Tcl or shell command. As the Tcl interface also offers access to HDL Companion's internal data structure, you can write your own Tcl scripts to generate reports or perform specific checks.

Design Flow

The tool flow wizard allows you to configure a complete flow of other tools (like simulation, synthesis and FPGA place and route) you want to use in your project. After running the wizard, buttons are present in the toolbar to start your tools. In combination with the toplevel marker HDL Companion can compile the HDL files with the selected simulator or generate synthesis scripts for the synthesis tool.

HDL Works

HDL Works B.V.
Maxwellstraat 19
P.O. Box 620, 6710 BP Ede
The Netherlands
Phone: +31 (0)318 642 022
Fax: +31 (0)318 636 482
E-mail: info@hdlworks.com
www.hdlworks.com

Features and Benefits

- HDL Companion
  - HDL Languages
    - VHDL, Verilog and mixed language
    - Syntax checking
    - Works on incomplete designs
    - Language sensitive editor/viewer
    - Detailed view of design objects
    - File / Library / Hierarchical views
    - Design metrics
  - Productivity
    - Up and running with your design in minutes
    - Extract design structure from source code in seconds
    - Relates between use and definition
    - Quickly navigate through your design
    - Automate recurring tasks
    - Version management and revison control
    - Extensive search functionality
    - Signal tracing through the whole hierarchy
  - Design flow and methodology
    - Fits in any tool flow
    - Distributed directories
    - Includes miscellaneous files (eg synthesis scripts or PDF doc.)
    - Tcl driven backend
    - Interfaces for major simulator and synthesis tools
    - Run all required tools from one environment

Language support

Analysis
VHDL87, VHDL93
Verilog, Verilog 2001

Editing
VHDL, Verilog, Java,
SystemC, Tcl, EDIF, Perl

EDA Vendor Support

Actel
Altera
Mentor Graphics
Model Technology
Synplicity
Aldec
Cadence
Silos
Synopsys
Xilinx

Operating Systems

PC	Windows 2000/XP
PC	RedHat Linux 7.3 or later
Sun SPARC	Solaris 2.7 or later