DSipher Design Solutions Pvt. Ltd.
206 Kristal Olivine, Outer Ring Road, Bellandur, Bangalore 560037, Karnataka, INDIA
Tel: +91-80-41643253 Email: contactus@dsipherdesign.com http://dsipherdesign.com

USER GUIDE

IPintentio

Table of Contents

1 Introduction

IPintentio is a web browser based environment that integrates Integrated Circuit – Digital Design flows along with a parallel software simulation methodology. An overview of the benefits and capabilities of IPintentio are outlined below:

Flexible and customizable EDA software-agnostic design flow

- Including open source simulator and coverage tool for front-end design
- Including a methodology and open source tools to design and simulate reference software for the said hardware

Platform-independence

- Web browser user interface

Program management capabilities, such as

- Project management utilities
- Selective permissions assignment
- Data integrity across members, teams and generations

Integrated Development Environment (IDE) utilities, such as

- File browser, code editor, terminal access
- Visual design browser

The subsequent sections will explain in detail how to use this environment.

2 Quick Start

- **Create a Project**

  This is the very first step.

  You must create a project to proceed. Project creation requires a project name (mandatory), design files (this can be done at any stage of the project), and contact email addresses of your collaborators (if any; they will be invited to register and collaborate). This information can be edited at any time. Note that some features are dependent on the information you provide (e.g. the design browser requires RTL files).

  Once a project is added it will appear in the existing projects list. You may edit its details at any point of time subsequently.

- **Invite collaborators**

  You may invite collaborators and assign appropriate privileges to each.

  Owner is assigned complete access to all project data. An owner can edit or delete the project and enjoys complete access to project files through the file browser and terminal interface.

  Member is assigned complete access to all project data. However, a member cannot edit or delete the project. Level of access to project files is identical to that of an Owner.

  Reviewer is assigned partial access to project data. A reviewer cannot edit or delete a project. Level of access to project files is restricted to read-only through the file browser.

- **Create a project plan**

  A project member with **owner** privileges may create and maintain a plan to track using the project planner utility.

- **Workspaces**

  Plan includes project creation, project editing and planning utilities

  Design includes tool execution, terminal, file editor and design browser utilities.

- **Tag files for RTL Browser**

  The RTL browser utility in the **Design** tab can be used to visualize the RTL files (uploaded or created online) in a project. The tagging step is required to identify RTL files in your file repository.

- **View / Edit files**

  You can create / view / edit a file using the text editor utility. It has a simple text edit mode (notepad-like) and a VI edit mode which simulates the most commonly used key-map of the VIM text editor.

- **Browse around in the terminal**

  The terminal utility emulates a LINUX command shell. A restricted set of LINUX commands are supported. Type *help* in the terminal for further information.

- **Simulate your design**

  You can simulate your design and testbench files using the Icarus Verilog simulator through the IPintentio design flow setup.

- **View coverage**

  You can view the coverage metrics of your verification testcases using the Covered coverage tool.

- **Generate Co-Simulation Model**
  You can create a model from your verilog component rtl files for the provided hardware-software Co-Simulation setup.

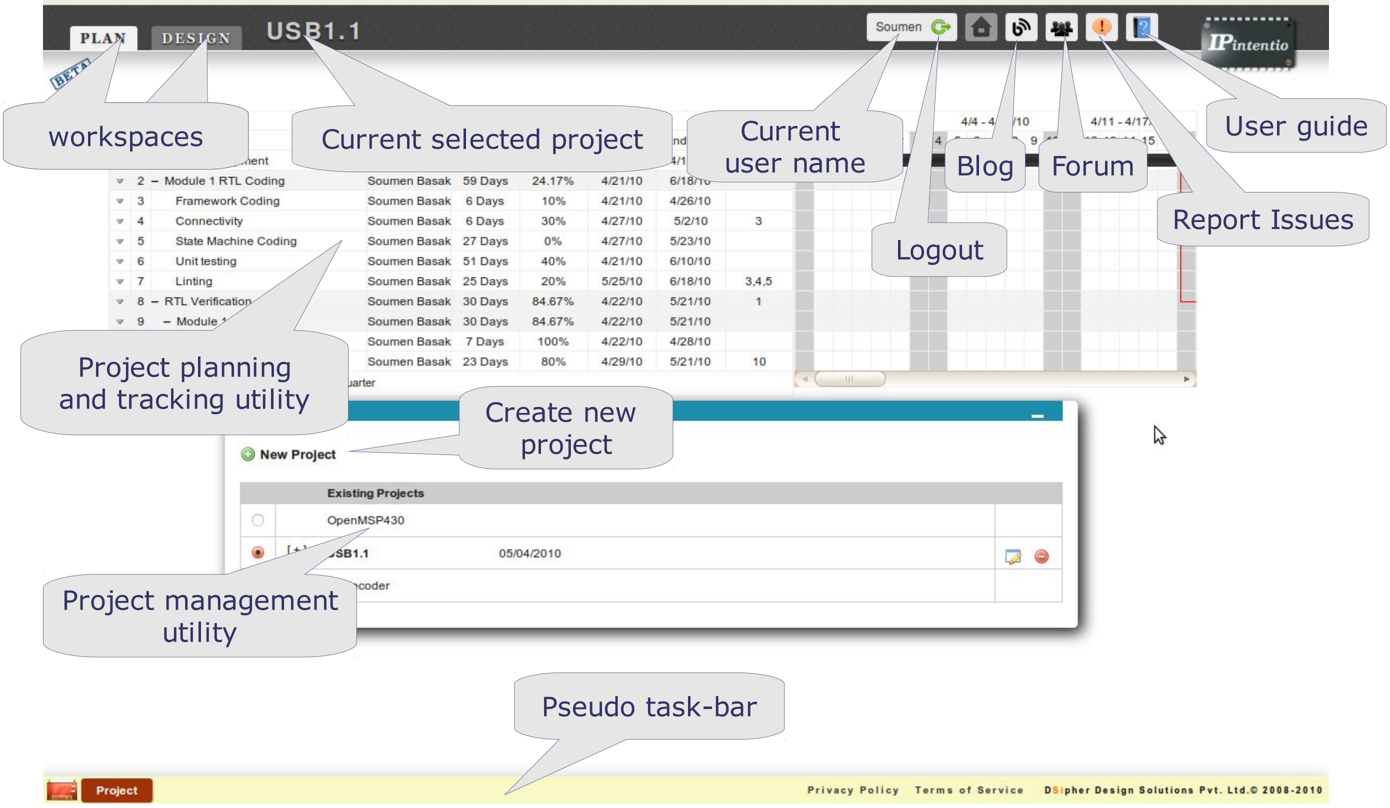
- **Software-Configuration**
  You can compile one or more C programs to create an image file to load and run on a simulated HW board.

- **Co-Simulation**
  You can load and run an image file on a chosen simulated hardware board through this step. You can also specify any generated models and their related configuration(s) to be a part of the simulated hardware board.

3 Plan

3.1 Overview

This includes project creation, project editing and planning utilities



		ment					4/1	
▾	2	– Module 1 RTL Coding	Soumen Basak	59 Days	24.17%	4/21/10	6/18/10	
▾	3	Framework Coding	Soumen Basak	6 Days	10%	4/21/10	4/26/10	
▾	4	Connectivity	Soumen Basak	6 Days	30%	4/27/10	5/2/10	3
▾	5	State Machine Coding	Soumen Basak	27 Days	0%	4/27/10	5/23/10	
▾	6	Unit testing	Soumen Basak	51 Days	40%	4/21/10	6/10/10	
▾	7	Linting	Soumen Basak	25 Days	20%	5/25/10	6/18/10	3,4,5
▾	8	– RTL Verification	Soumen Basak	30 Days	84.67%	4/22/10	5/21/10	1
▾	9	– Module 1	Soumen Basak	30 Days	84.67%	4/22/10	5/21/10	
			Soumen Basak	7 Days	100%	4/22/10	4/28/10	
			Soumen Basak	23 Days	80%	4/29/10	5/21/10	10
		uarter						

New Project

	Existing Projects		
○	OpenMSP430		
●	[+] USB1.1	05/04/2010	
	coder		

Fig 1 : 'Plan' workspace screen-shot

3.2 *Project Management*

3.2.1 Create a new project

Click on the 'New Project' button to enter details of the new project to be created. Details to be entered:

- **Project Name**: It should be less than 25 characters, with no whitespace and can contain alphabets and integers and '_' (underscore). This can be changed later.

- **Upload File**: Files must be compressed. (Zip, Compress, Gzip and Bzip)

- **Start Date:** Project creation date will be chosen if this field is left blank. This can be edited later

- **Digital Design Flow**: Choose the Digital Design flow to be used in the project. Currently only one flow is available

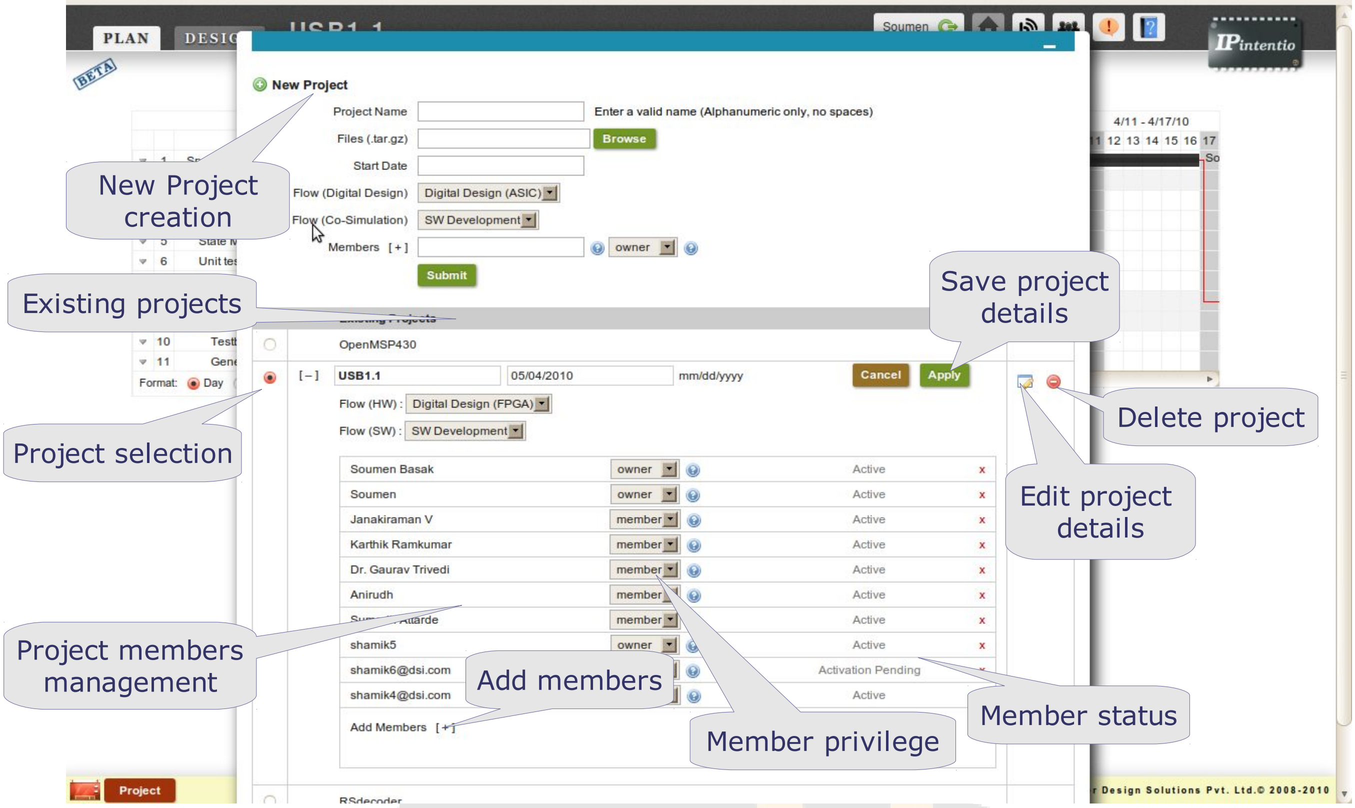
- **Co-simulation Flow**: Choose the flow to be used in the project. Currently only one flow is available.

- **Members**: Specify email address and privilege level per collaborator. Members can be added later

Once the project is added it should appear in the existing projects list and selected as the current working project.

3.2.2 Edit existing project

Users with **Owner** can edit a project.



New Project

Project Name Enter a valid name (Alphanumeric only, no spaces)

Files (.tar.gz) Browse

Start Date

Flow (Digital Design) Digital Design (ASIC)

Flow (Co-Simulation) SW Development

Members [+] owner

Submit

Existing Projects

○ OpenMSP430

● [–] USB1.1 05/04/2010 mm/dd/yyyy Cancel Apply

Flow (HW) : Digital Design (FPGA)

Flow (SW) : SW Development

Soumen Basak	owner	Active	x
Soumen	owner	Active	x
Janakiraman V	member	Active	x
Karthik Ramkumar	member	Active	x
Dr. Gaurav Trivedi	member	Active	x
Anirudh	member	Active	x
[illegible] Atarde	member	Active	x
shamik5	owner	Active	x
shamik6@dsi.com		Activation Pending	x
shamik4@dsi.com		Active	

Add Members [+]

○ RSdecoder

Fig 9 : Project management features screen-shot

3.2.3 Collaborating

You may invite collaborators and assign appropriate privileges to each.

3.2.3.1 *Add / Remove project members*

A user with **Owner** privileges, can add or remove members to or from the project. Added members, if already registered, will receive an intimation email. There is no further action required on his part and the new project is displayed in his existing project table. If not registered, a pre-approved activated email will be sent to be specified email address. The new member must activate his account to access this project.

3.2.3.2 *Privilege types*

The privilege characteristic of a member determines the permissions granted to him / her for a project.

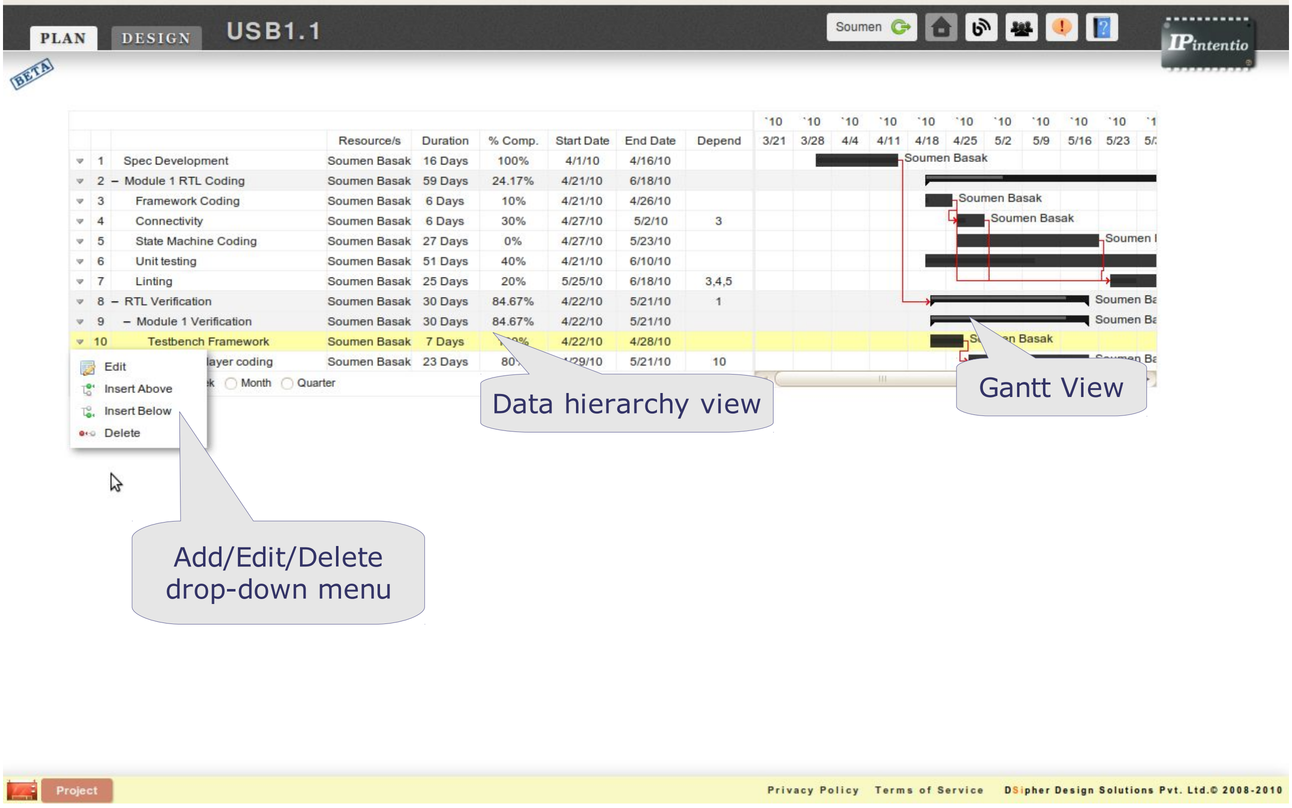
- **Owner** is assigned complete access to all project data. An owner can edit or delete the project and enjoys complete access to project files through the file browser and terminal interface.
- **Member** is assigned complete access to all project data. However, a member cannot edit or delete the project. Level of access to project files is identical to that of an Owner.
- **Reviewer** is assigned partial access to project data. A reviewer cannot edit or delete a project. Level of access to project files is restricted to read-only through the file browser.

3.3 Project Planning / Scheduling

This utility is provided to enable users to plan and track the progress of a project. Currently we provide basic features needed to accomplish project tracking.

These are:

- Add/Edit/Delete tasks and milestones
- Associate project members to tasks and milestones
- Associate start date and end date
- Specify percentage completion
- Task categorization and hierarchy
- Specify dependencies between tasks
- Daily, weekly, monthly, quarterly Gantt-chart visualization of tasks and milestones



		Resource/s	Duration	% Comp.	Start Date	End Date	Depend
1	Spec Development	Soumen Basak	16 Days	100%	4/1/10	4/16/10	
2	– Module 1 RTL Coding	Soumen Basak	59 Days	24.17%	4/21/10	6/18/10	
3	Framework Coding	Soumen Basak	6 Days	10%	4/21/10	4/26/10	
4	Connectivity	Soumen Basak	6 Days	30%	4/27/10	5/2/10	3
5	State Machine Coding	Soumen Basak	27 Days	0%	4/27/10	5/23/10	
6	Unit testing	Soumen Basak	51 Days	40%	4/21/10	6/10/10	
7	Linting	Soumen Basak	25 Days	20%	5/25/10	6/18/10	3,4,5
8	– RTL Verification	Soumen Basak	30 Days	84.67%	4/22/10	5/21/10	1
9	– Module 1 Verification	Soumen Basak	30 Days	84.67%	4/22/10	5/21/10	
10	Testbench Framework	Soumen Basak	7 Days	[illegible]	4/22/10	4/28/10	
	layer coding	Soumen Basak	23 Days	80	/29/10	5/21/10	10

Data hierarchy view

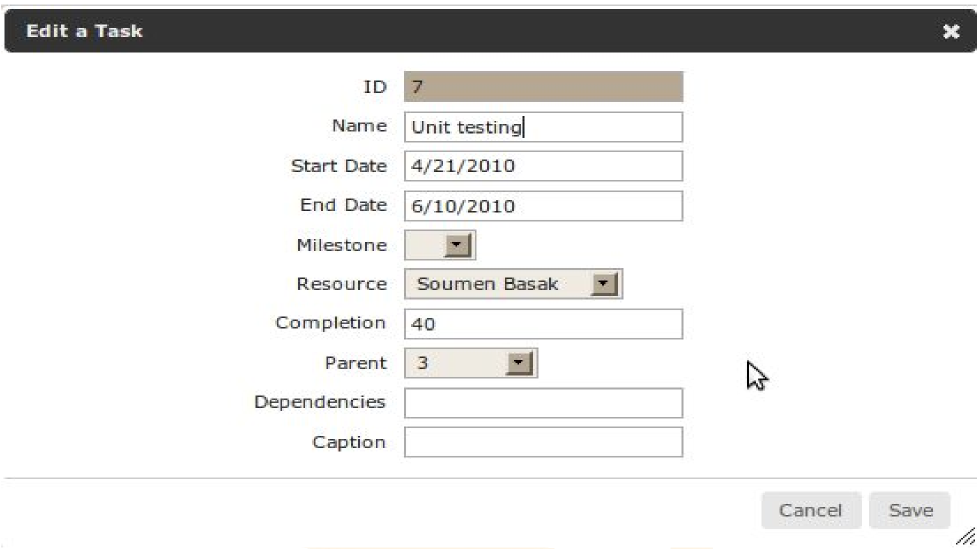
Gantt View

Add/Edit/Delete drop-down menu

Fig 9 : Project management features screen-shot

3.3.1 Tasks & Milestones

Tasks and milestones can be added, edited and/or deleted from a drop-down menu which can be accessed by clicking on the down-arrow at the start of each row. A dialog box opens which contains the relevant fields that need to be filled in or edited by the user.



Edit a Task

ID	7
Name	Unit testing
Start Date	4/21/2010
End Date	6/10/2010
Milestone	
Resource	Soumen Basak
Completion	40
Parent	3
Dependencies	
Caption	

Fig 10 : Edit task dialog screen-shot

3.3.2 Recommendations

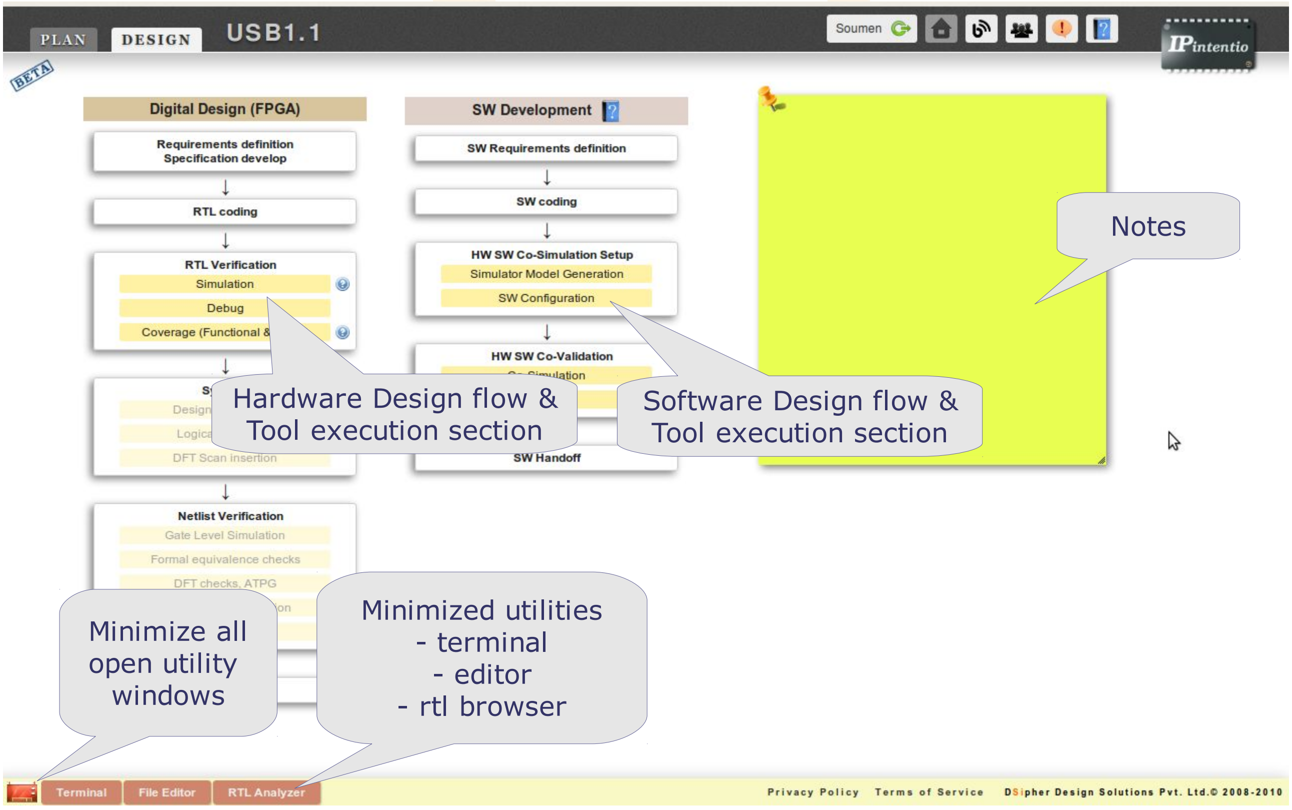
The following actions may lead to unexpected behavior in the project management utility

- Not specifying a start and end date.
- Deleting tasks which are parents of other tasks.
- Deleting tasks on which other tasks are dependent.
- Specifying invalid percentage numbers.
- Specifying task ids of undefined tasks as parents or dependencies.

4 Design

4.1 Overview

This includes tool execution, terminal, file editor and design browser utilities.



Notes

Hardware Design flow &
Tool execution section

Software Design flow &
Tool execution section

Minimize all
open utility
windows

Minimized utilities
- terminal
- editor
- rtl browser

Fig 2 : 'Design' workspace screen-shot

4.2 Utilities

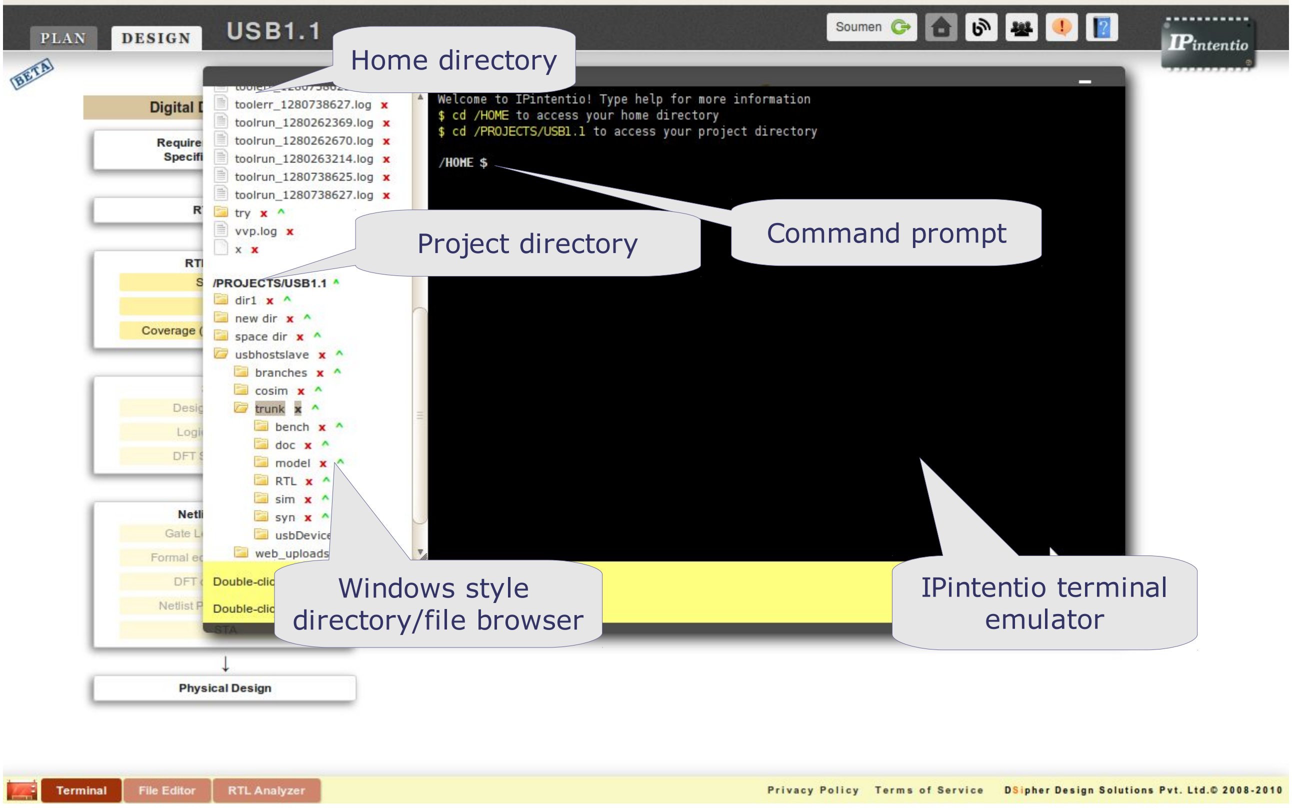
These are similar to common desktop applications that a user needs to design effectively. The utilities currently provided are

1. Terminal & Directory/File browser
2. A Code editor with a subset of vi key bindings.
3. RTL hierarchy browser

Each utility has its own window which is minimized by default.

4.2.1 Terminal

This utility comprises a LINUX terminal emulator and a Windows-like directory/file browser.



```
Welcome to IPintentio! Type help for more information
$ cd /HOME to access your home directory
$ cd /PROJECTS/USB1.1 to access your project directory

/HOME $
```

Home directory

Command prompt

Project directory

Windows style directory/file browser

IPintentio terminal emulator

Fig 3 : Terminal features screen-shot

4.2.1.1 Terminal Emulator

You can use this as a command shell inside your work area. A set of LINUX shell commands that are necessary for design is provided. In addition, some standard utilities and specific scripts can be invoked using this interface. Certain commands are not provided or are restricted for security purposes.

For more details type 'help' in the terminal command prompt.

4.2.1.2 Directory/File Browser

Files and directories in the user's HOME directory (/HOME) and the selected project directory (/PROJECTS/<projectname>) are listed here. This browser provides the following features:

- File and directory browsing capabilities.
- Option to remove a file or directory (x).
- Option to view/edit a text file in the *Editor* utility by double clicking on a listed file.
- Option to download simulation dump files (v). Only VCD, LXT, LXT2 extensions for dump files are supported.
- Option to upload to directories (^).

Delete and Upload features in the project area are governed by user privileges.

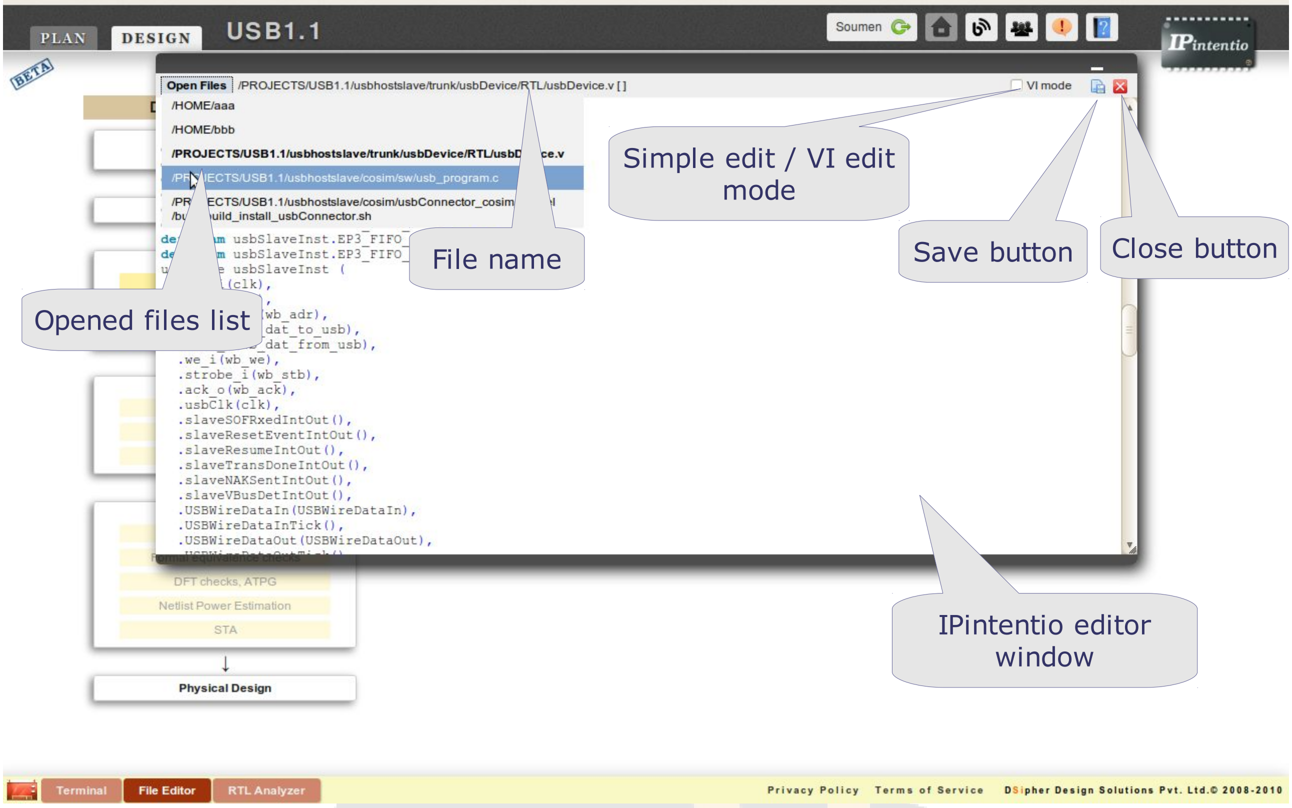
4.2.2 Editor

You can create / view / edit a file using the text editor utility. It has a simple text edit mode (notepad-like) and a VI edit mode which simulates the most commonly used key-map of the VIM text editor. It includes syntax highlighting for verilog (.v) and C, C++ files (.c and .cpp).

Multiple files can be opened in the editor. User can switch between open files from the 'Open Files' drop-down menu.

A file can be loaded in the editor in 3 ways

1. By double-clicking a listed file in the directory/file browser
2. By running the gvim/vim/vi command in the terminal command prompt.
3. By right-clicking on the respective module in the RTL browser (opens the respective verilog file, in the editor).

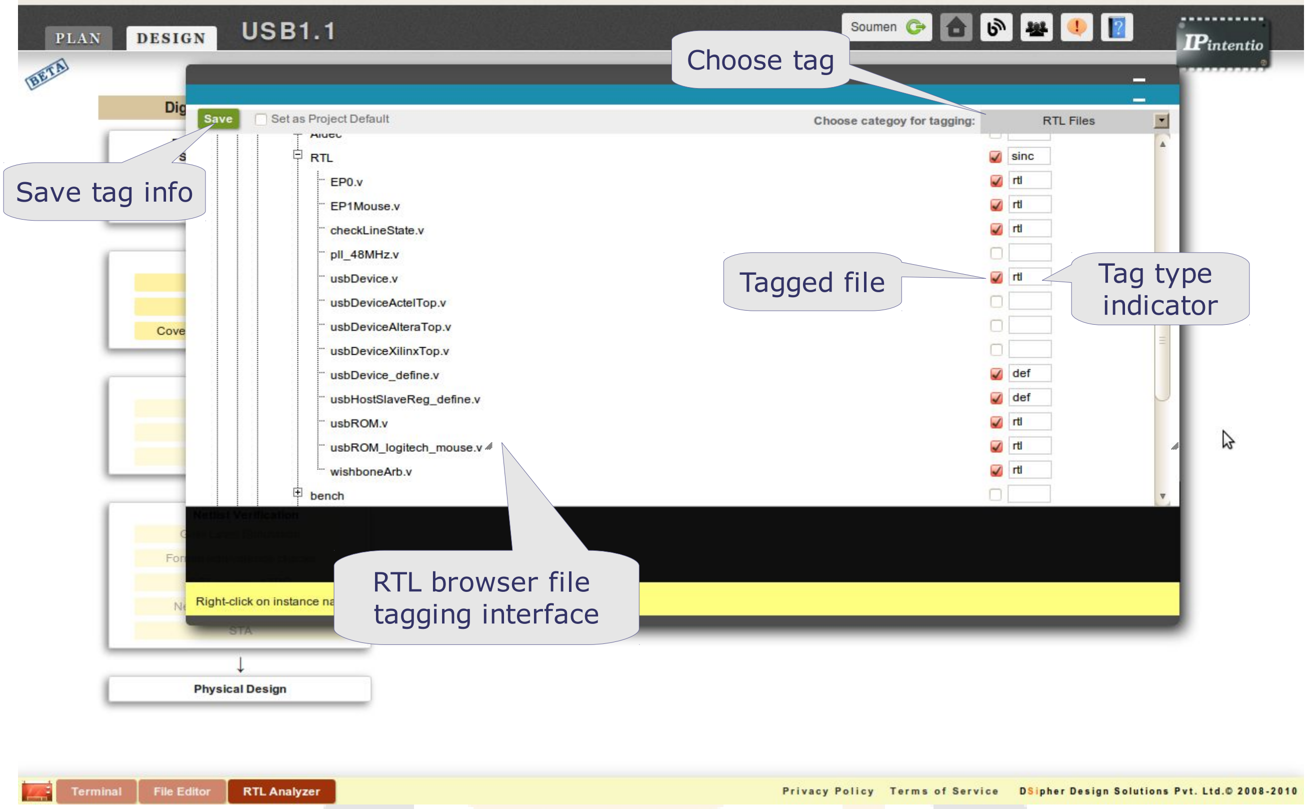

PLAN
DESIGN
USB1.1
Soumen
BETA
IPintentio
Open Files
/PROJECTS/USB1.1/usbhostslave/trunk/usbDevice/RTL/usbDevice.v [ ]
VI mode
/HOME/aaa
/HOME/bbb
/PROJECTS/USB1.1/usbhostslave/trunk/usbDevice/RTL/usbD ce.v
/PR ECTS/USB1.1/usbhostslave/cosim/sw/usb_program.c
/PR ECTS/USB1.1/usbhostslave/cosim/usbConnector_cosim el
/bu uild_install_usbConnector.sh
Simple edit / VI edit mode
File name
Save button
Close button
Opened files list
IPintentio editor window
DFT checks, ATPG
Netlist Power Estimation
STA
Physical Design
Terminal
File Editor
RTL Analyzer
Privacy Policy
Terms of Service
DSipher Design Solutions Pvt. Ltd.© 2008-2010

```
de     m usbSlaveInst.EP3_FIFO_
de     m usbSlaveInst.EP3_FIFO_
u      e usbSlaveInst (
         (clk),
          ,
          (wb_adr),
          dat_to_usb),
          dat_from_usb),
  .we_i(wb_we),
  .strobe_i(wb_stb),
  .ack_o(wb_ack),
  .usbClk(clk),
  .slaveSOFRxedIntOut(),
  .slaveResetEventIntOut(),
  .slaveResumeIntOut(),
  .slaveTransDoneIntOut(),
  .slaveNAKSentIntOut(),
  .slaveVBusDetIntOut(),
  .USBWireDataIn(USBWireDataIn),
  .USBWireDataInTick(),
  .USBWireDataOut(USBWireDataOut),
```

Fig 4 : Editor features screen-shot

4.2.3 RTL Browser

The RTL browser provides a visualization of the design in the selected project as also a utility to tag files.



Choose tag

Save tag info

Tagged file

Tag type indicator

RTL browser file tagging interface

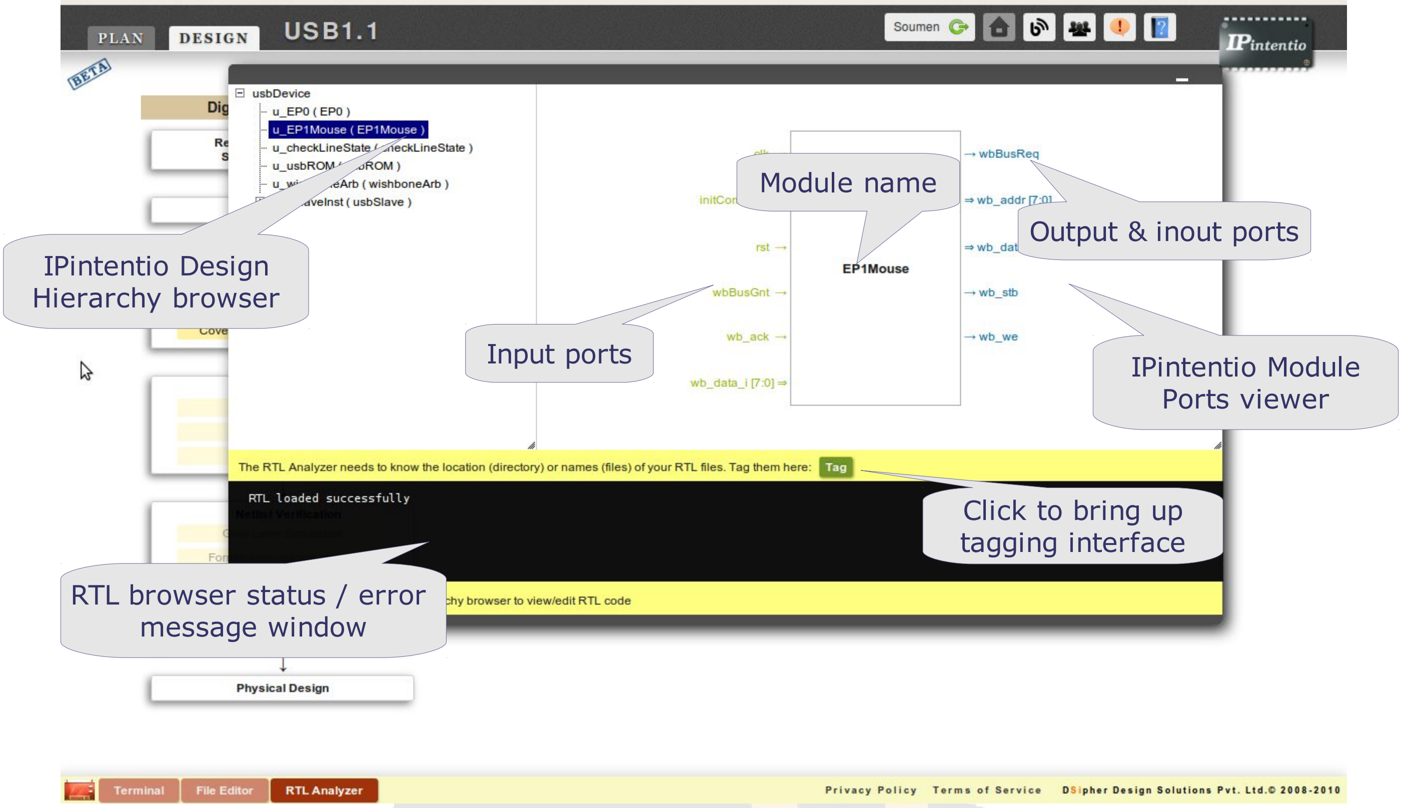
□ Set as Project Default

Choose categoy for tagging: RTL Files

RTL	☑	sinc
EP0.v	☑	rtl
EP1Mouse.v	☑	rtl
checkLineState.v	☑	rtl
pll_48MHz.v	□	
usbDevice.v	☑	rtl
usbDeviceActelTop.v	□	
usbDeviceAlteraTop.v	□	
usbDeviceXilinxTop.v	□	
usbDevice_define.v	☑	def
usbHostSlaveReg_define.v	☑	def
usbROM.v	☑	rtl
usbROM_logitech_mouse.v	☑	rtl
wishboneArb.v	☑	rtl
bench	□	

Right-click on instance na

Fig 5 : RTL browser tagging interface screen-shot



The RTL Analyzer needs to know the location (directory) or names (files) of your RTL files. Tag them here:

```
RTL loaded successfully
```

...chy browser to view/edit RTL code

IPintentio Design Hierarchy browser

Module name

Output & inout ports

Input ports

IPintentio Module Ports viewer

Click to bring up tagging interface

RTL browser status / error message window

Fig 5 : RTL browser features screen-shot

The RTL browser supports only Verilog 2001 HDL. A subset of testbench constructs are supported and it can generate hierarchy and port information of testbenches correctly in most cases.

It consists of 4 parts

1. Design Hierarchy
2. Module ports view
3. File tagging interface
4. Status / Message area

4.2.3.1 Design Hierarchy

This shows the module hierarchy of the whole design. You can browse the design hierarchy, click on a particular module to view its ports in the *Module ports viewer* and also open the file that defines that module in the *Editor* with a right-click.

4.2.3.2 *Module ports view*

This gives a view of the module ports (inputs, outputs, inouts) for a given module in a hierarchy.

4.2.3.3 *File tagging interface*

The tagging step is required to identify RTL files in your file repository. There are 4 different types of tags that are accepted as inputs

1. RTL files (rtl) – files contain the RTL code. This is mandatory.

2. Include directories (sinc) – directories in which to look for files which have been included in RTL using the `*include* directive. This is optional and is needed only if the `*include* directive is used

3. RTL defines files (def) – files which contain `*define* directives. This is optional and is needed only if `*define* directives are present in separate files.

4. RTL files to exclude (xcld) – Files which should not be considered by RTL browser. This is needed when one needs to include netlist instances of modules without including the library cells. This is optional.

4.2.3.4 Status / Message area

This area shows the status of the RTL browser engine after analysis of the tagged files specified by the user. You can view these messages to correct any errors in tagging or coding in the RTL files to properly visualize your design.

4.3 *Design Flow*

The Digital Design and Co-simulation flows are sections provided to execute EDA software on your design. The flows can be configured to provide a custom set of tool options at each flow step.

IPi Lite uses the following EDA software:

- Icarus Verilog version 0.9.2

- covered-0.7.8

- Verilator – 3.803

- Verilog-Perl – 3.301

- SID

- eCos – 3.0

The figure below depicts a standard digital design flow. It is broken up into sub-flows and further into constituent flow-steps

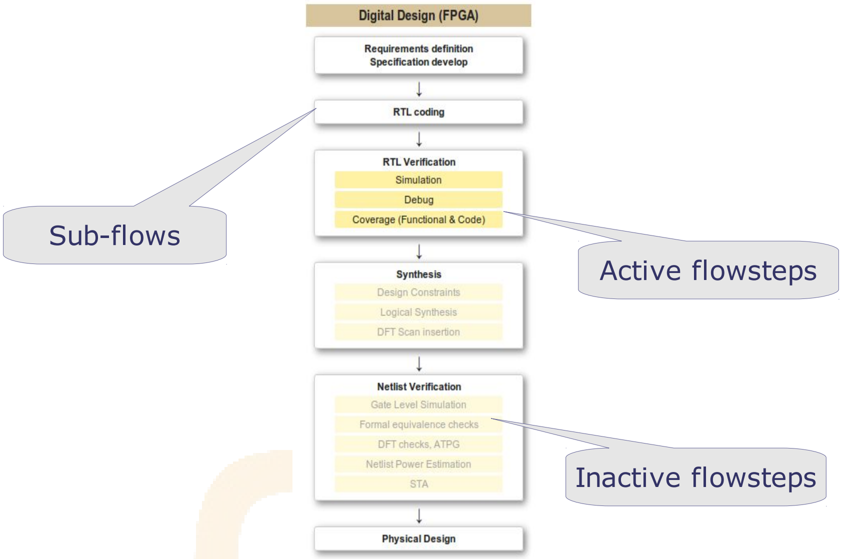


Fig 6 : Flow screen-shot

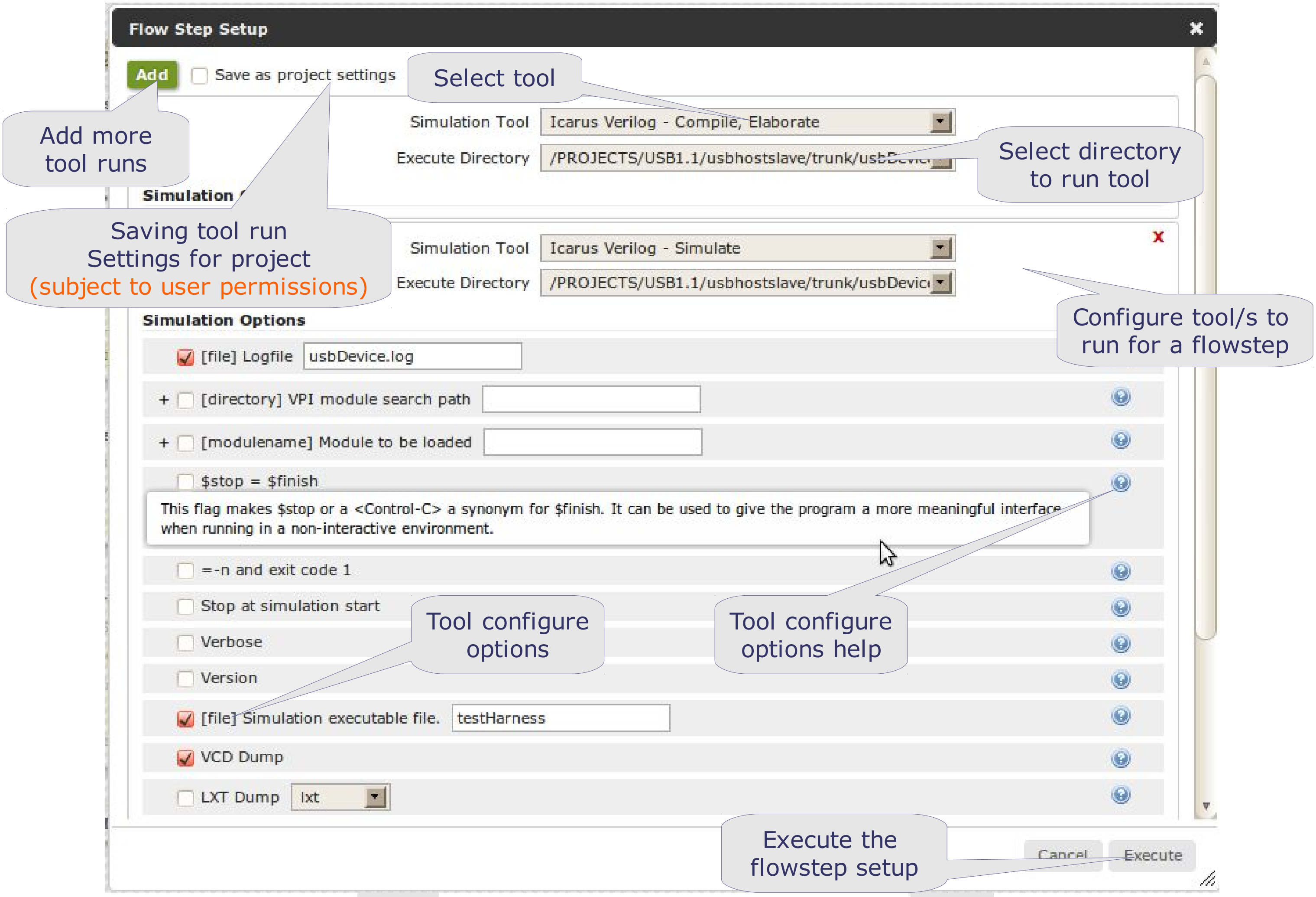
4.3.1 Tool Execution

Clicking on an active flow-step displays the tool run dialog. This dialog lists the tools that can be run in that flow-step and the execution directory. You can specify multiple tools or the same tool with different options to be run in a single flow-step.

Upon selecting a tool, its options are displayed. These can be viewed by clicking the 'Options' link (e.g. Simulation Options). Further details of each option can be viewed by clicking the help icon ?.

Options specified in a particular toolrun are retained. Users with **Owner** privileges save a particular set of options for use across user on the same project. However a user can override project settings in for a particular toolrun.

It is advised to execute commands in one flowstep at a time.



☐ Save as project settings

Simulation Tool	Icarus Verilog - Compile, Elaborate
Execute Directory	/PROJECTS/USB1.1/usbhostslave/trunk/usbDevic

Simulation

Simulation Tool	Icarus Verilog - Simulate
Execute Directory	/PROJECTS/USB1.1/usbhostslave/trunk/usbDevic

Simulation Options

- ☑ [file] Logfile usbDevice.log
- + ☐ [directory] VPI module search path
- + ☐ [modulename] Module to be loaded
- ☐ $stop = $finish

This flag makes $stop or a <Control-C> a synonym for $finish. It can be used to give the program a more meaningful interface when running in a non-interactive environment.

- ☐ =-n and exit code 1
- ☐ Stop at simulation start
- ☐ Verbose
- ☐ Version
- ☑ [file] Simulation executable file. testHarness
- ☑ VCD Dump
- ☐ LXT Dump lxt

Select tool

Add more tool runs

Select directory to run tool

Saving tool run Settings for project (subject to user permissions)

Configure tool/s to run for a flowstep

Tool configure options

Tool configure options help

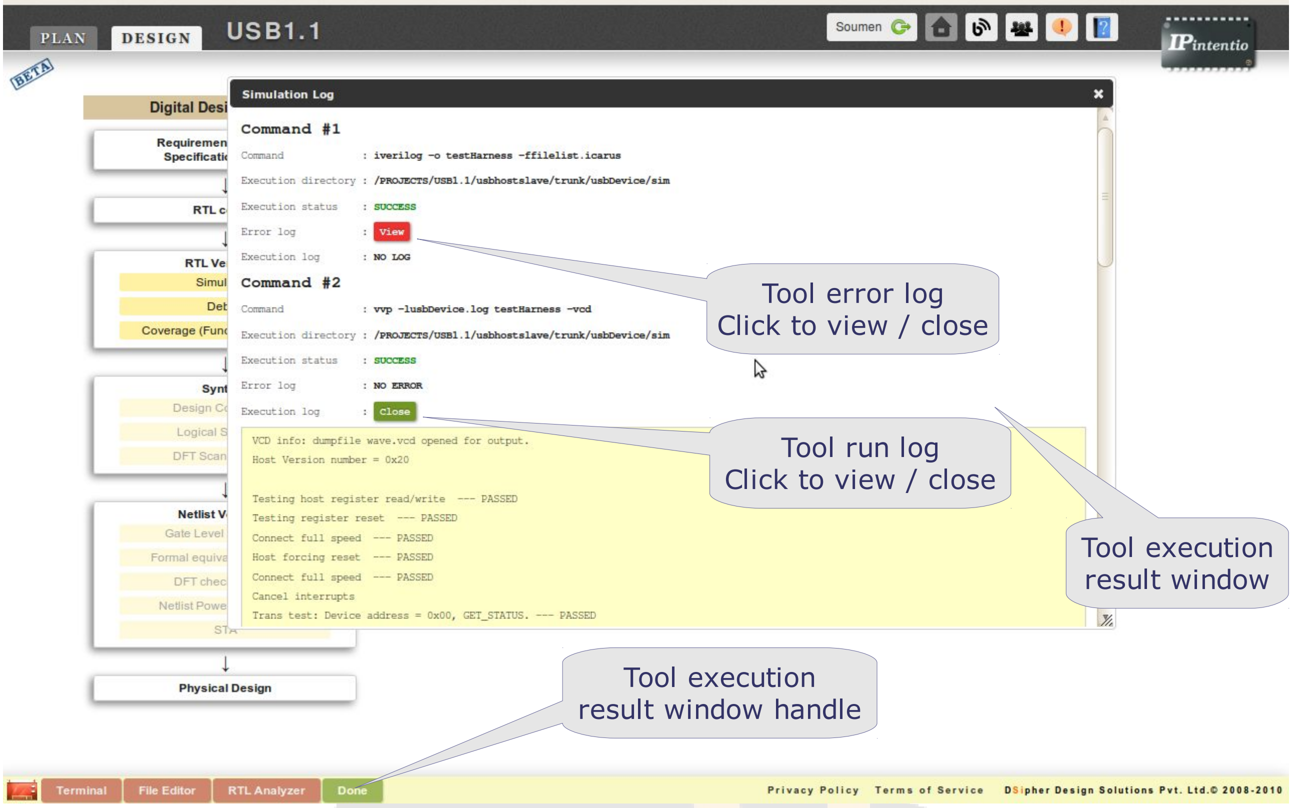
Execute the flowstep setup

Fig 7 : Tool run dialog screen-shot

4.3.2 Execution Logs/Reports

Tool execution output is displayed in a dialog once the execution is complete. This dialog provides the following information:

- Complete tool execution command
- Tool execution result.
- The tool execution output is stored in files in the command execution directory. These logs or reports generated as a file can be accessed using the Directory/File browser or the Terminal command prompt.



```
Command #1
Command             : iverilog -o testHarness -ffilelist.icarus
Execution directory : /PROJECTS/USB1.1/usbhostslave/trunk/usbDevice/sim
Execution status    : SUCCESS
Error log           : View
Execution log       : NO LOG
Command #2
Command             : vvp -lusbDevice.log testHarness -vcd
Execution directory : /PROJECTS/USB1.1/usbhostslave/trunk/usbDevice/sim
Execution status    : SUCCESS
Error log           : NO ERROR
Execution log       : Close
  VCD info: dumpfile wave.vcd opened for output.
  Host Version number = 0x20

  Testing host register read/write  --- PASSED
  Testing register reset  --- PASSED
  Connect full speed  --- PASSED
  Host forcing reset  --- PASSED
  Connect full speed  --- PASSED
  Cancel interrupts
  Trans test: Device address = 0x00, GET_STATUS. --- PASSED
```

Tool error log
Click to view / close

Tool run log
Click to view / close

Tool execution
result window

Tool execution
result window handle

Fig 8 : Tool execution log screen-shot