Getting Started Using Aldec's Active-HDL

This guide will give you a short tutorial in using the project mode of Active-HDL. This tutorial is broken down into the following sections

1. Part 1: Compiling a Design
2. Part 2: Simulating a Design
3. Transitioning to the Next Lab
4. Setting Simulator Resolution
5. Directory Structures and Running Scripts

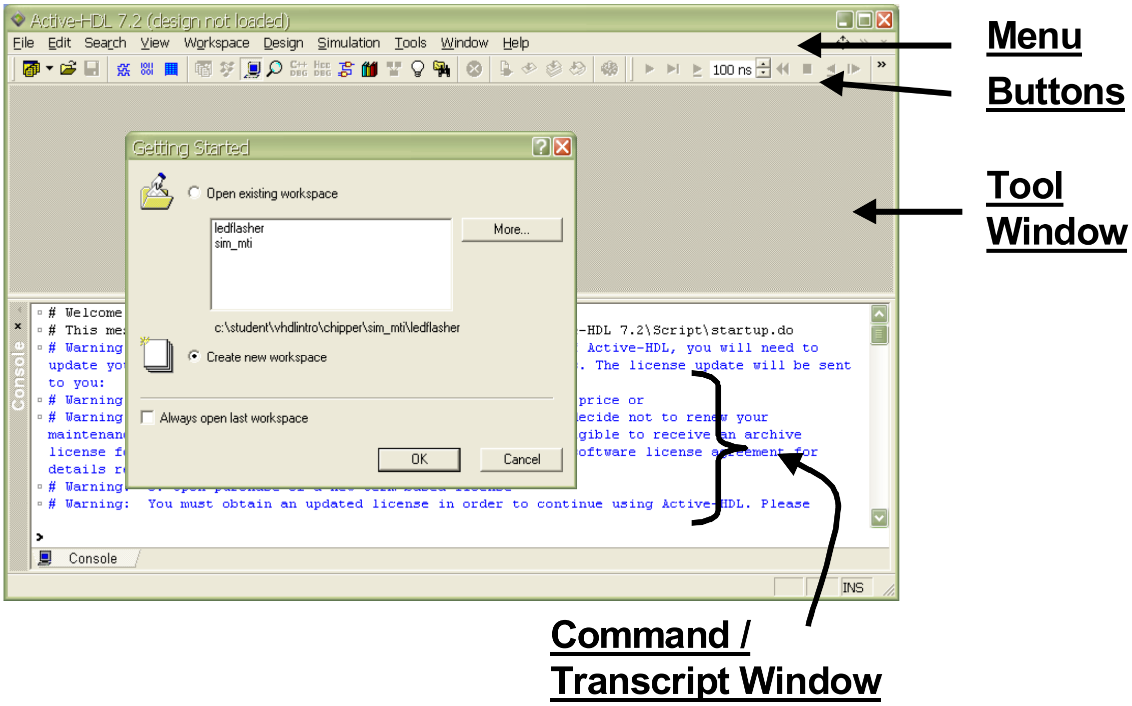
1 Part 1: Compiling a Design

1.1 Code Your Design

Code your design per the lab directions.

1.2 Start Active-HDL

Start Active-HDL by double clicking on the Active-HDL Icon (windows).


Active-HDL 7.2 (design not loaded)
File Edit Search View Workspace Design Simulation Tools Window Help
100 ns
Getting Started
Open existing workspace
More...
Create new workspace
Always open last workspace
OK
Cancel
Console
INS
Menu
Buttons
Tool Window
Command / Transcript Window

ledflasher
sim_mti

c:\student\vhdlintro\chipper\sim_mti\ledflasher

○ Open existing workspace

● Create new workspace

□ Always open last workspace

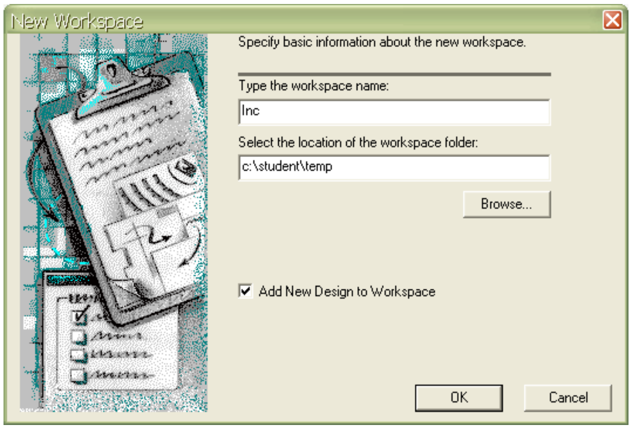
```
# Welcome                       
# This me                       -HDL 7.2\Script\startup.do
# Warning                        Active-HDL, you will need to
update yo                       . The license update will be sent
to you:
# Warning                       price or
# Warning                       ecide not to renew your
maintenan                       gible to receive an archive
license f                       oftware license agreement for
details r
# Warning:
# Warning:  You must obtain an updated license in order to continue using Active-HDL. Please
>
```

When you open Active-HDL, a dialog box will open to open a workspace. If you have already created workspace, select and open it. Otherwise, select create a new workspace and work through the steps in the next section.

1.3 Creating a Workspace

If you closed the dialog box from the previous section, you can get to this step by selecting the menu command, “File > New > Workspace”.

In the first dialog box, specify workspace name to match the design name (Inc) and specify the workspace folder to be put in the directory “VhdlIntro/Chipper/vhdl_src”. Make sure the box, “Add new Design to Workspace” is checked. This is shown below.



New Workspace

Specify basic information about the new workspace.

Type the workspace name: Inc

Select the location of the workspace folder: c:\student\temp

☑ Add New Design to Workspace

In the second dialog, check the box marked, “Add Existing Resource File” and press next. In the third dialog, browse to your VHDL source directory (for lab 1 it is “VhdlIntro/Chipper/vhdl_src”) and select the design files (for the first part of lab 1 it is just Inc.vhd) and press next. In the fourth dialog, press next. In the fifth dialog, specify the design name (Inc) and press next. In the last dialog, press finish.

In this step you created the workspace, added a design (Inc) and compiled it. If you had errors, read the next section to help understand them, skip the “Adding Files” section (until you create your testbench), and then recompile your design using the steps in “Compile a Design”.

1.4 Understanding Compile Errors

The key to understanding error messages is to realize that the message is produced after the compiler gets lost. In the example below, the entity Adder8 (not the lab design) has a ‘;’ after the last port declaration. Most often this means that the error message will be generated on line 41. This is common, so make sure when you get an error message to read the specific error message and realize the root cause may be something done on the previous line that causes the compiler to “detect” the error on the current line.

```
36  -- not the lab design
37  entity Adder8 is
38    port (
39      A, B : In  std_logic_vector(7 downto 0);
40      Y    : Out std_logic_vector(7 downto 0);  -- error
41    );
42  end Adder8 ;
```

Note that you can double click on an error message and editor will take you to the line in error.

1.5 Adding Files

As your project grows, you may need to add files. To do this, use the menu command: "Design > Add Files to Design".

1.6 Compile a Design

After correcting errors, recompile a design with the menu command: "Design > Compiles All".

After adding files, compile a design with the menu command: "Design > Compiles All with File Reorder".

2 Simulating a Design

2.1 Code Your Testbench

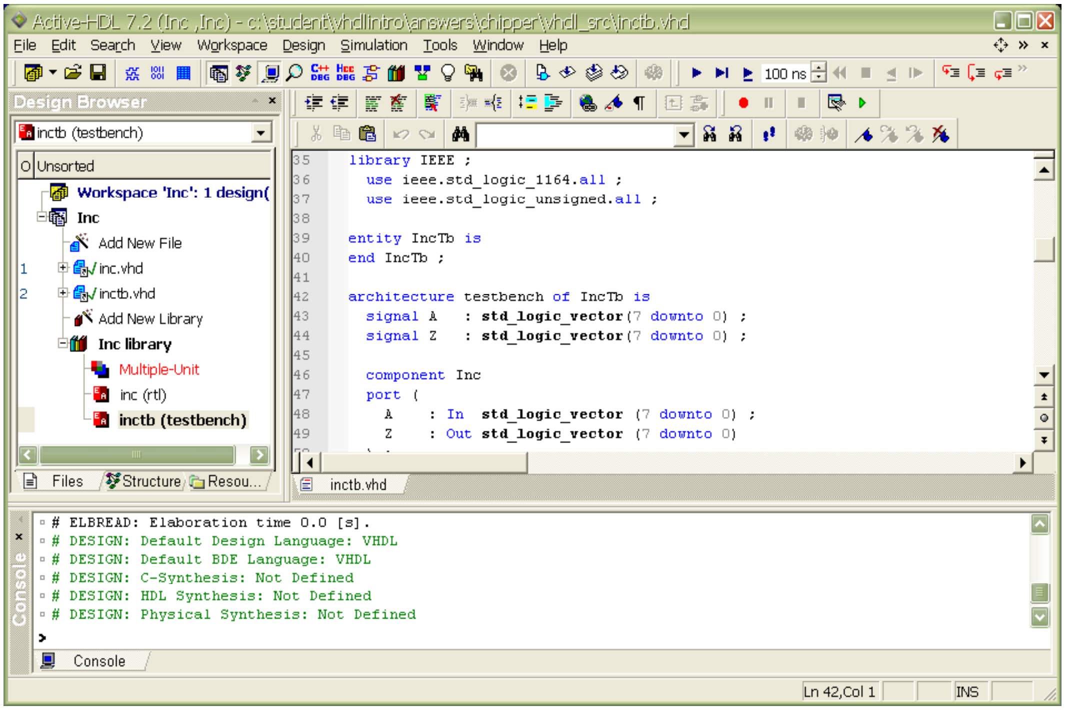
Code the testbench per lab directions.

2.2 Add Testbench and Compile

Add IncTb.vhd to your design using the steps from "Adding Files" and then compile IncTb.vhd using the setps from "Compile a Design". Make sure to use "Compiles All with File Reorder" as the testbench needs to be compiled last.

2.3 Load the Testbench

To run a simulation, we need to select the testbench in the library, set it as the top-level, and then initialize the simulation. In the Design Browser window, expand the library (Inc) and select the testbench (inctb). This is shown in the picture below. Right click on the testbench (inctb) and select "Set as Top-Level". Now select the menu command, "Simulation > Initialize Simulation".

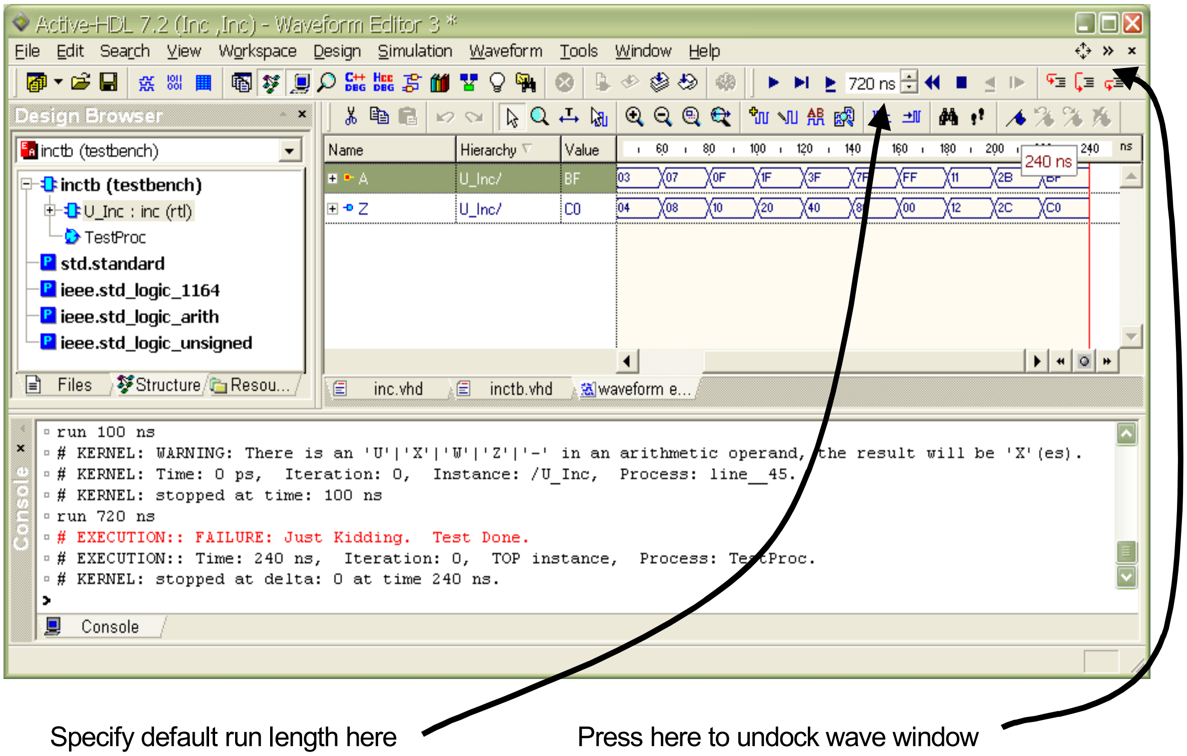


```
library IEEE ;
  use ieee.std_logic_1164.all ;
  use ieee.std_logic_unsigned.all ;

entity IncTb is
end IncTb ;

architecture testbench of IncTb is
  signal A    : std_logic_vector(7 downto 0) ;
  signal Z    : std_logic_vector(7 downto 0) ;

  component Inc
  port (
    A    : In  std_logic_vector (7 downto 0) ;
    Z    : Out std_logic_vector (7 downto 0)
```

```
# ELBREAD: Elaboration time 0.0 [s].
# DESIGN: Default Design Language: VHDL
# DESIGN: Default BDE Language: VHDL
# DESIGN: C-Synthesis: Not Defined
# DESIGN: HDL Synthesis: Not Defined
# DESIGN: Physical Synthesis: Not Defined
>
```

2.4 Display Waveforms

Display the waveforms before running the simulation and after loading the testbench. To display waveforms, select the design under test (inc) in the “structure” tab of the design browser, right-click the mouse, and select “Add to Waveform”.



```
run 100 ns
# KERNEL: WARNING: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
# KERNEL: Time: 0 ps,  Iteration: 0,  Instance: /U_Inc,  Process: line__45.
# KERNEL: stopped at time: 100 ns
run 720 ns
# EXECUTION:: FAILURE: Just Kidding.  Test Done.
# EXECUTION:: Time: 240 ns,  Iteration: 0,  TOP instance,  Process: TestProc.
# KERNEL: stopped at delta: 0 at time 240 ns.
>
```

Specify default run length here

Press here to undock wave window

Once a waveform window is available, you may drag entities to the wave window from the Design Browser structure tab into the waveform window.

2.5 Run the Simulation

For lab 1, you will be running your design for 720 ns. To do this, change the default run length (see picture above) to 720 ns (type it in the box). Now select “Simulate > Run For”.

Note that it is quite normal to see warning messages at the start of the simulation. Most messages that occur before a design is initialized and/or reset can be ignored. For example, the following warning results from the A input being initialized to “UUUUUUUU” at time 0:

```
# KERNEL: WARNING: There is an 'U'|'X'|'W'|'Z'|'-' in an
arithmetic operand, the result will be 'X'(es).
# KERNEL: Time: 0 ps,  Iteration: 0,  Instance: /U_Inc,  . . .
```

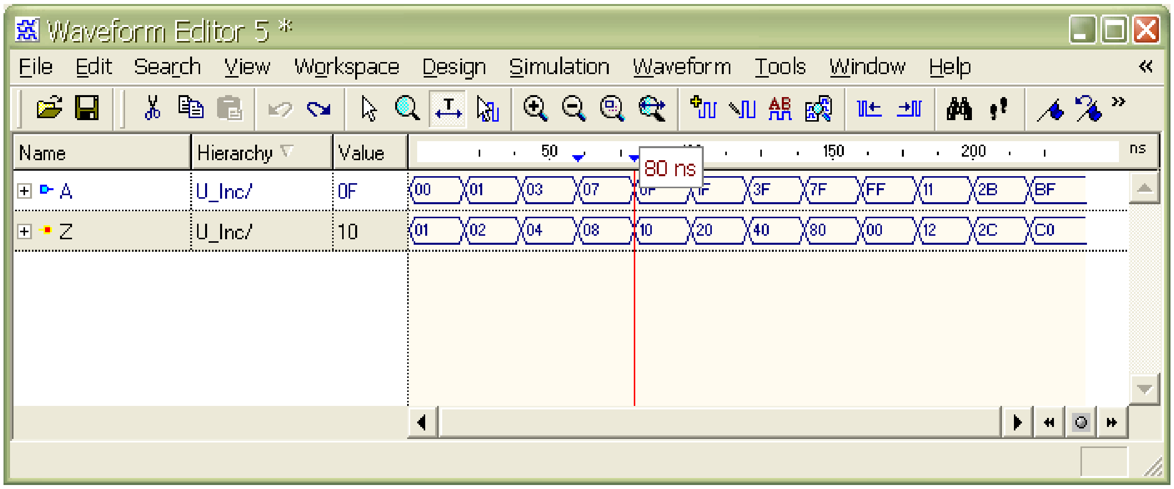
Note that if you forgot to display your waveforms before running the simulation or find a bug during simulation, you will need to re-run the simulation. See the next section for details.

2.6 Recompiling and Rerunning a Simulation

VHDL designs are required to be compiled bottom-up. In Active-HDL, the tool re-arranged the compile order for you when you used “Design > Compiles All with File Reorder”. Once you have done this, use the “Design > Compiles All” command to rebuild the design hierarchy. Then re-run the simulation.

2.7 Using the Waveform Window

All commands described in this section are done with the waveform window undocked. To undock the wave window press on the undock button (see diagram in the Displaying Waveforms section – it looks like “>>”). The waveform window menus appear as shown below.



Name	Hierarchy	Value
A	U_Inc/	0F
Z	U_Inc/	10

2.7.1 Displaying Waveforms after Running a Simulation

If you display waveforms after running a simulation and want to see the waveforms from the beginning of time, you will need to restart your design (using the restart command) and rerun the simulation.

2.7.2 Zooming

Zoom In, Out, and Fit is available with “View > Zoom >“ menu. They are also available from the button bar.

2.7.3 Selecting Signals

Select one signal by clicking the left mouse button on its name.
To select multiple signals,

1. Select one signal and then press shift-left mouse button to select a range
2. Select one signal and then add more signals by pressing ctrl-left mouse button.

2.7.4 Moving Signals

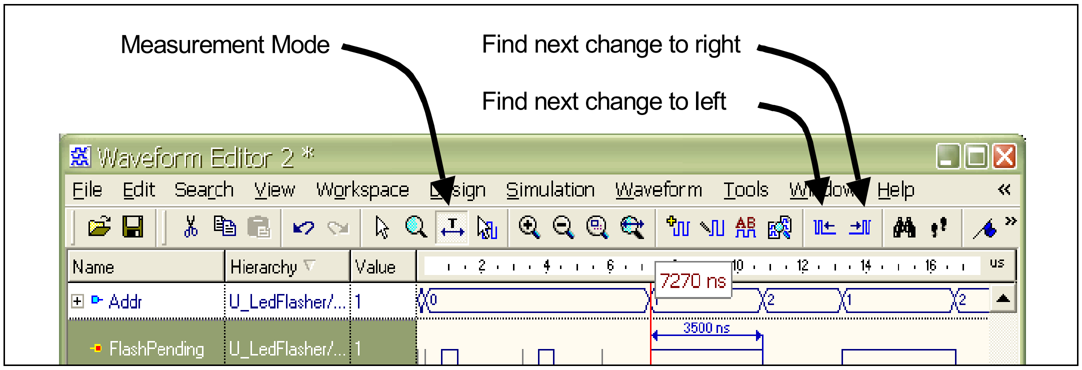
Select signal(s) and drag to new location in wave window.

2.7.5 Radix

The radix of the selected signal (or signals) can be specified using the "Edit > Properties" menu command.

2.7.6 Using Cursors

To accurately place a cursor, use the find next change symbols (shown below) to move the active cursor to the next change of the selected signal. Alternately dragging a cursor over an edge of the selected signal will cause the cursor to try to snap to the edge.



To measure the time between two points, select a signal and then use the measurement mode.

3 Transitioning to the Next Lab

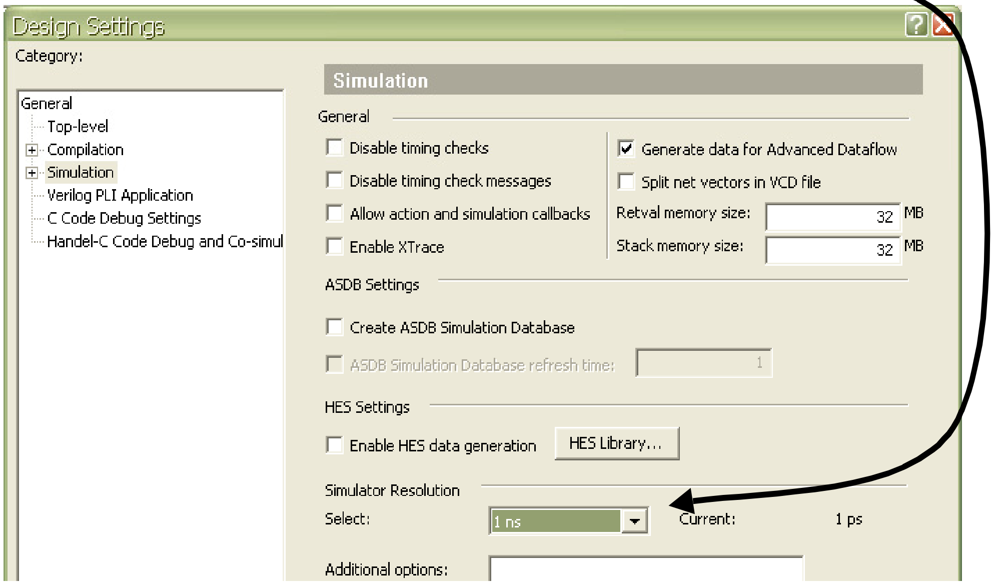
If you are starting a new project, start with the steps described in "Creating a Workspace".

If you are continuing on the same project (like LedCtrl and LedFlasher …), you can simply add files to the current project using the steps described in "Adding Files".

4 [Optional] Setting Simulator Time Resolution

In Aldec, the simulator resolution is set to Auto by default. To change the simulator resolution, use the "Design > Settings" menu command and then select the "Simulation" option (as shown below. And set the simulator resolution in the box shown below.

Simulation Resolution: Note this is not the default run length.



Simulation

General

☐ Disable timing checks

☐ Disable timing check messages

☐ Allow action and simulation callbacks

☐ Enable XTrace

☑ Generate data for Advanced Dataflow

☐ Split net vectors in VCD file

Retval memory size: 32 MB

Stack memory size: 32 MB

ASDB Settings

☐ Create ASDB Simulation Database

☐ ASDB Simulation Database refresh time: 1

HES Settings

☐ Enable HES data generation

Simulator Resolution

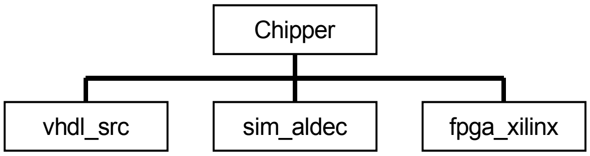
Select: 1 ns

Current: 1 ps

Additional options:

5 Directory Structures & Running Scripts

Currently we are running the simulator out of the VHDL source directory (vhdl_src). An alternate way to organize a project is to have a separate directory for source files and separate directories for the simulator and synthesis tools. The following directory structure is being used for these labs:



Exit out of the simulator. In the directory vhdl_src, delete any file that does not have a *.vhd extension. Delete the subdirectory work.

Start the simulator. Use the "tools > Execute Macro" menu command to run the script file named **TbLedFlasher_lab6.tcl** that is in the directory "VhdlIntro/Chipper/sim_aldec". Note only *.do files are visible by default and you will have to use the dialog box to enable seeing either *.tcl files or all files (*.*). Since Aldec has a project style compile, the main use of a script like this is to initialize a project the first time (such as when a new member joins a project or the project is un-archived).