Using Lattice's ispLever Starter Software

This document is a "how to" on producing JEDEC files for GALs using Verilog.

Leif Ilvedson
April 28, 2002

Revision 1:
Joe Barta
July 8, 2002

Revision 2:
James K. Peckol
September 8, 2005

Overview

The following is a procedure on the basic use of Lattice's ispLever Starter Software in conjunction with Synplicity's Synplify software.

Scope

The scope of the following procedure is to produce a JEDEC file for the target PLD, the GAL22V10D-15LP, using Verilog. This tutorial will leave you with the ability to produce JEDECs for either the GAL22V10 or GAL16V8 along with their different versions (referring to their suffixes: A, B, C, D, -15LP, -25LP, etc).

A BCD-to-7-segment-decoder implemented using a GAL16V8D will be used to demonstrate the design process.

Estimated Time to Completion: 20 minutes

Big Picture

You will first use Lattice's ispLSS (ispLever Starter Software) to set up your project then import your Verilog code (.v). Then you'll use the softwareto produce a .jed (JEDEC) file. The JEDEC file is what will be used to program your GAL.

Strategy

If you make no pin assignments, ispLSS will automatically make pin assignments for you. This is an expletive-saver, especially if your design is complex and uses most of the pins. The pin assignments ispLSS makes are optimal for your design. Use this to your advantage:

1) When you are finished writing your Verilog code, let ispLSS automatically assign pins for you.
2) Take those pin assignments and hard-code them. That way, if you need to make any changes in your code, the pin assignments won't change and therefore you won't have to do any rewiring.

Understanding the Procedure Protocol

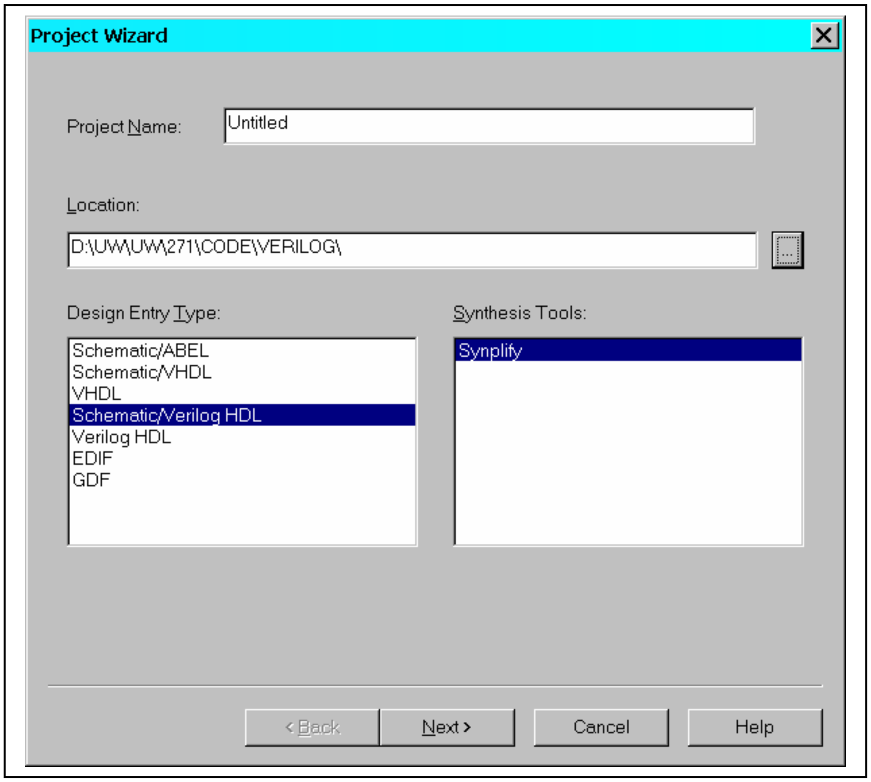
The following protocol may be seen: go to **File/New Project**. This implies that one would click on the **File** menu, then on **New Project** button. Duh.

Procedure

1. Start the ispLever Starter Software.

2. Go to **File/New Project**. This will start the *Project Wizard*

3. In the *Design Entry Type* menu, select **Schematic/Verilog HDL** as in Figure 1.



Project Name: Untitled

Location: D:\UW\UW\271\CODE\VERILOG\

Design Entry Type:

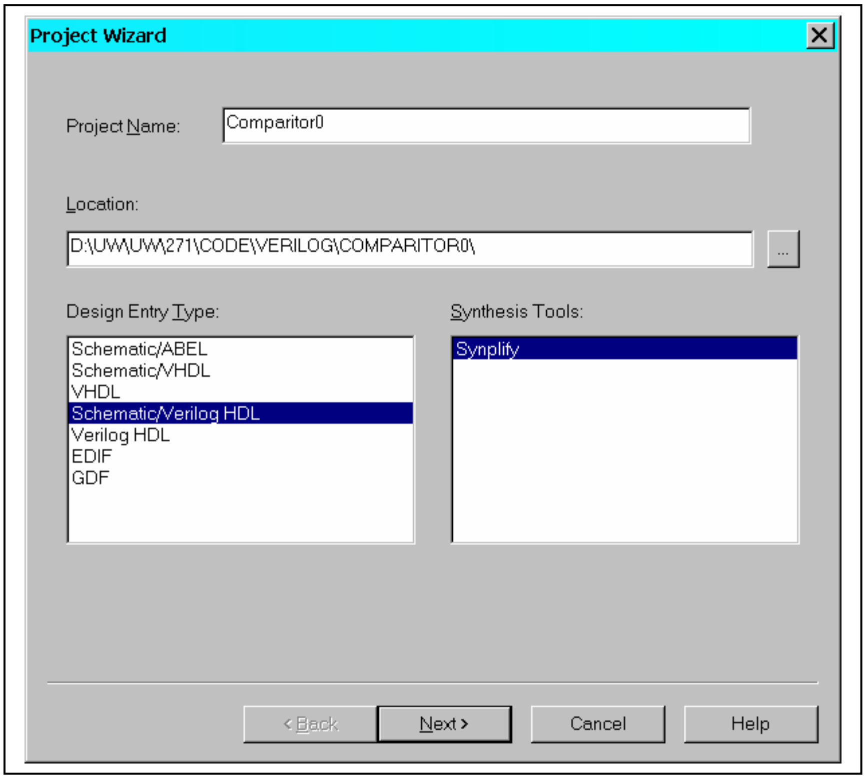
- Schematic/ABEL
- Schematic/VHDL
- VHDL
- Schematic/Verilog HDL
- Verilog HDL
- EDIF
- GDF

Synthesis Tools: Synplify

Figure 1.

4. Give the project a title by selecting the *Project Name* block of the Project Wizard window as shown in Figure 2. Enter “Comparitor0”.

5. Give the project a location by selecting the *Location* block of the Project Wizard window as shown in Figure 2. Enter or browse to the desired location. Click **Next**.



Project Wizard

Project Name: Comparitor0

Location: D:\UW\UW\271\CODE\VERILOG\COMPARITOR0\

Design Entry Type: Schematic/Verilog HDL

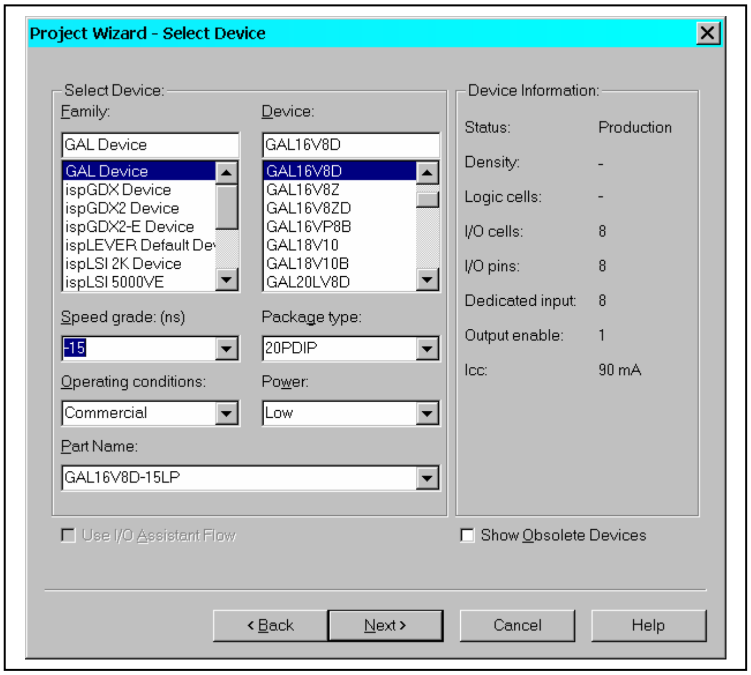
Synthesis Tools: Synplify

Figure 2.

6. Now the target device needs to be selected. *Under Select Device Family*, scroll up to **GAL Device** in the left pane of the Project Wizard window.

   In the *Select Device* window (Figure 3), select the appropriate choice from the drop-down menu. In this example, we use GAL16V8D. Then select the *Speed grade* and *Package type.*

   Click **Next**.



Project Wizard - Select Device

Select Device:

Family: GAL Device

Device: GAL16V8D

Speed grade: (ns) -15

Package type: 20PDIP

Operating conditions: Commercial

Power: Low

Part Name: GAL16V8D-15LP

Device Information:

Status:	Production
Density:	-
Logic cells:	-
I/O cells:	8
I/O pins:	8
Dedicated input:	8
Output enable:	1
Icc:	90 mA

Figure 3.

7. Select the Add Source button and browse to the appropriate *.v file.

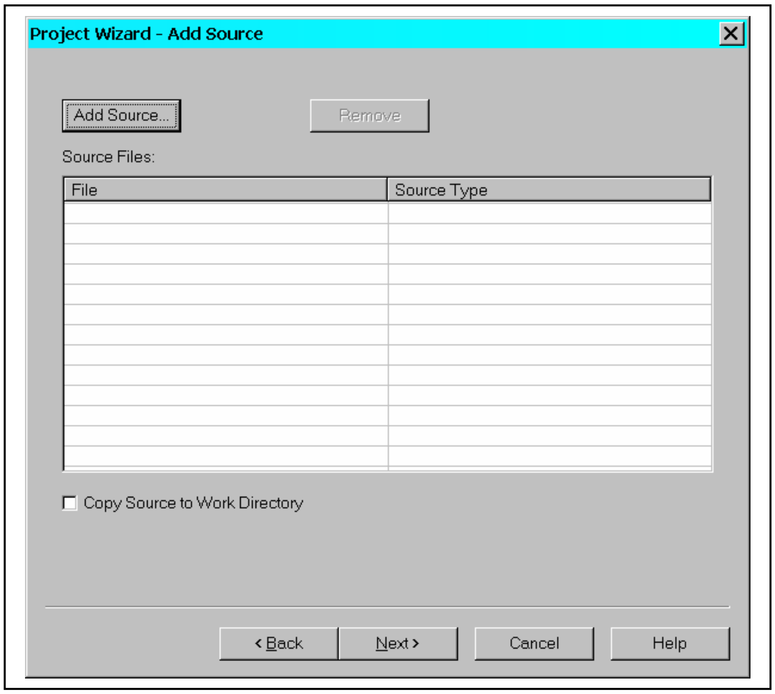
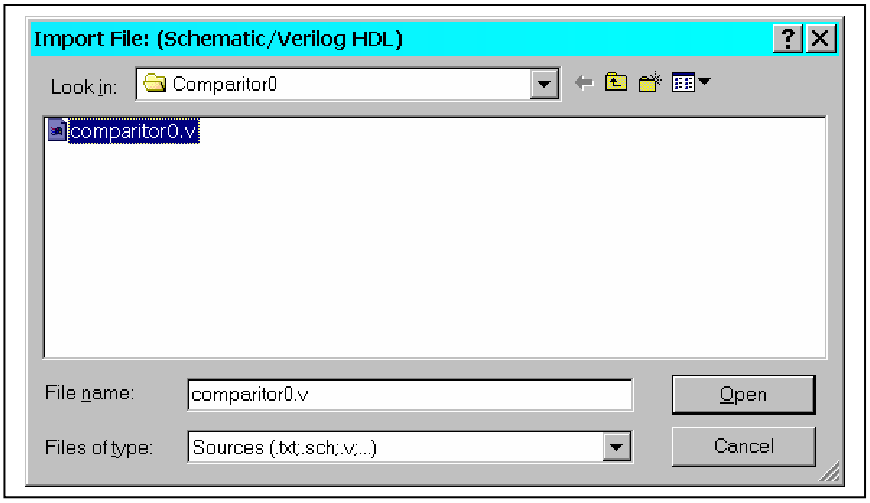


Figure 4.

8. Select the desired file and select *Open* as in Figure 5



Look in: Comparitor0

comparitor0.v

File name: comparitor0.v

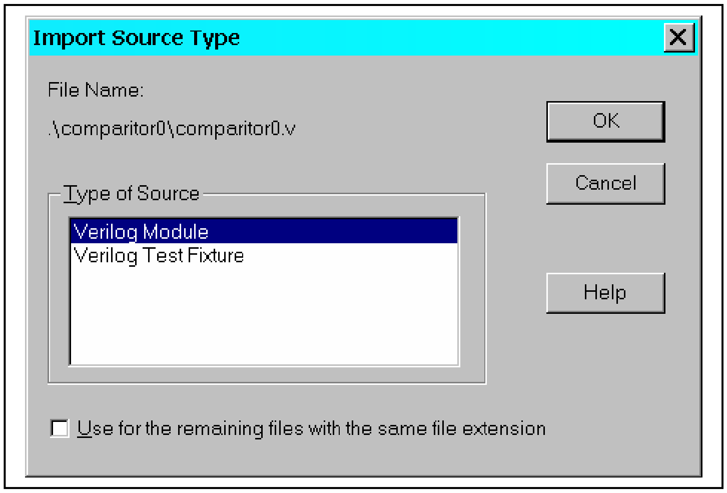
Files of type: Sources (.txt;.sch;.v;...)

Figure 5.

The comparator code appears in the following listing,

```
/*************************************************************
*  Title:                   Sample Verilog source code A Simple Comparitor
*  Model type:              Gate level
*  Description:             This Verilog source code simulates all input conditions
*                           of a greater than comparison
*  Programmer:              Your name
*  Date:                    Date typed
*************************************************************/


// Define the Comparitor module

module Comparitor(greater, a, b, c, d);

//  within the modules, wires are implied…we can put them in if we want to

     output    greater;       // Ouputs: greater
     input      a, b, c, d;   // Inputs: to compare, ab and cd


     // This is a structural model

     //   We're building a gate level model...  with propagation delay
     //   YOU DO NOT HAVE TO TYPE THESE COMMENT LINES.

     //  Gate instantiation format:
     //  name-of-module #(prop. delay) assigned-name (outputs-list, inputs_list)

     and       and1(term0, a, notC);
     and       and2(term1, b, notC, notD);
     and       and3(term2, a, b, notD);

     not       inv0(notC, c);
     not       inv1(notD,d);

     or        or1(greater, term0, term1, term2);

endmodule     //close Comparitor module
```

9. Select *Verilog Module* when the *Import Source Type* window pops open as in Figure 6. Select **OK**



File Name:

.\comparitor0\comparitor0.v

Type of Source

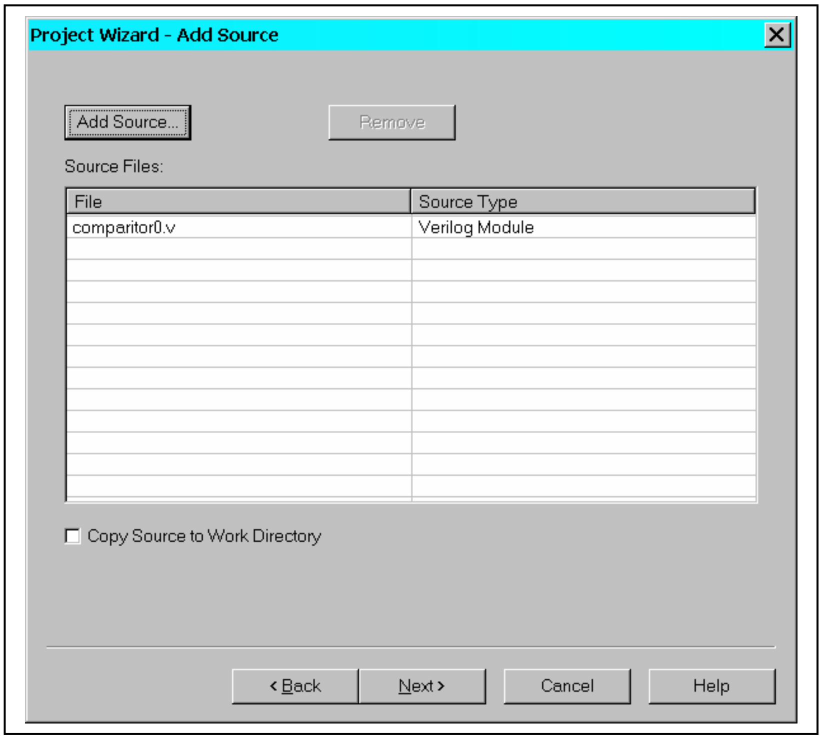
Verilog Module

Verilog Test Fixture

□ Use for the remaining files with the same file extension

Figure 6.

10. Click **Next**.



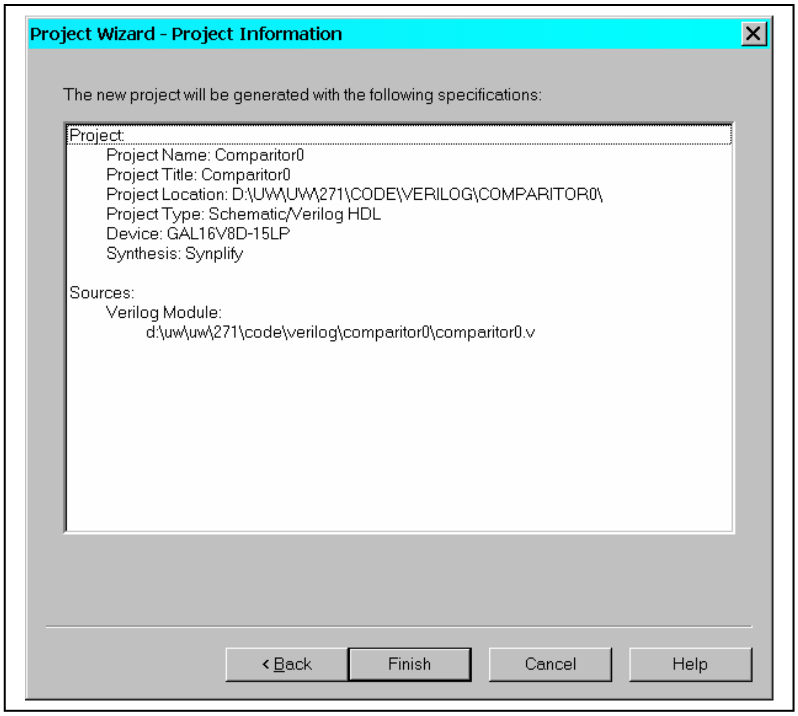
Source Files:

File	Source Type
comparitor0.v	Verilog Module

□ Copy Source to Work Directory

Figure 7.

11. Click **Finish**.

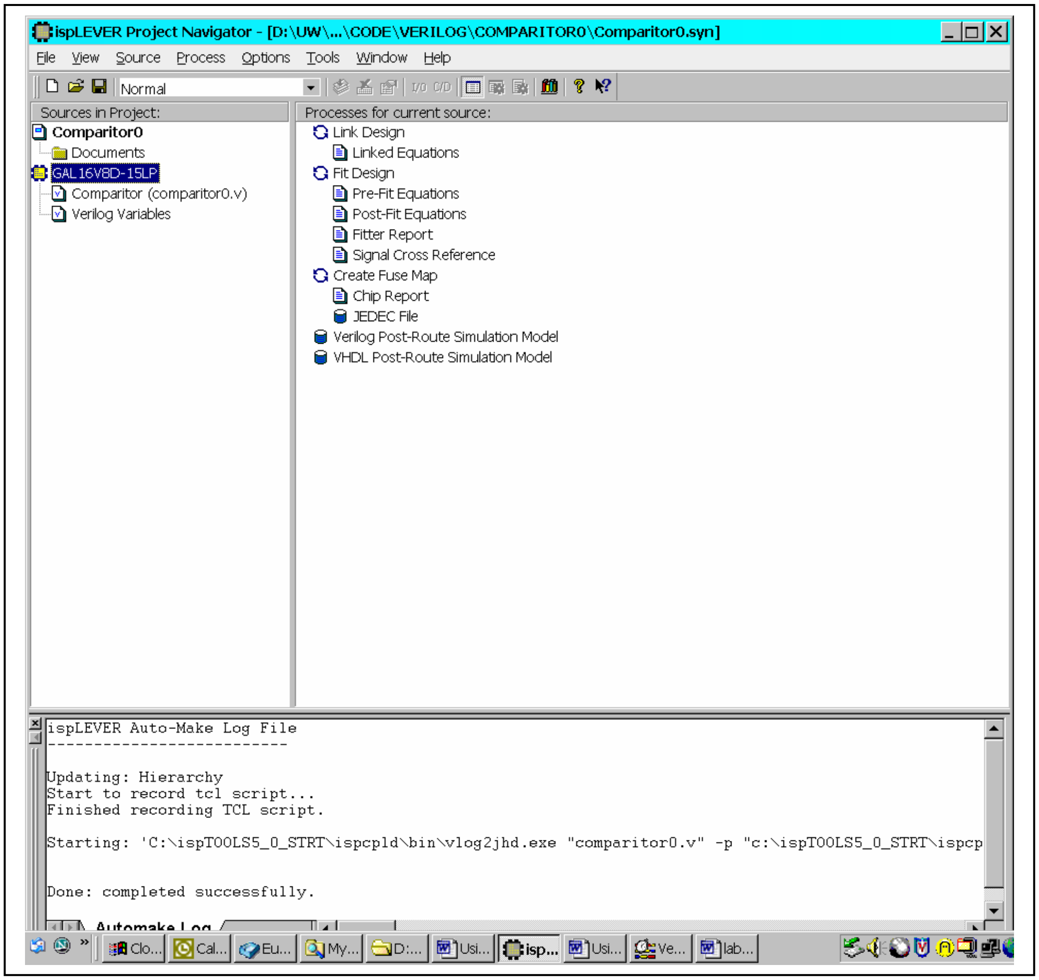


The new project will be generated with the following specifications:

```
Project:
        Project Name: Comparitor0
        Project Title: Comparitor0
        Project Location: D:\UW\UW\271\CODE\VERILOG\COMPARITOR0\
        Project Type: Schematic/Verilog HDL
        Device: GAL16V8D-15LP
        Synthesis: Synplify

Sources:
        Verilog Module:
                d:\uw\uw\271\code\verilog\comparitor0\comparitor0.v
```

Figure 8.

12. The Project Wizard exits to the Project Navigator screen as in Figure 9.


ispLEVER Project Navigator - [D:\UW\...\CODE\VERILOG\COMPARITOR0\Comparitor0.syn]
File View Source Process Options Tools Window Help
Normal
Sources in Project:
Comparitor0
Documents
GAL16V8D-15LP
Comparitor (comparitor0.v)
Verilog Variables
Processes for current source:
Link Design
Linked Equations
Fit Design
Pre-Fit Equations
Post-Fit Equations
Fitter Report
Signal Cross Reference
Create Fuse Map
Chip Report
JEDEC File
Verilog Post-Route Simulation Model
VHDL Post-Route Simulation Model
Automake Log
Clo... Cal... Eu... My... D:... Usi... isp... Usi... Ve... lab...

```
ispLEVER Auto-Make Log File
-------------------------

Updating: Hierarchy
Start to record tcl script...
Finished recording TCL script.

Starting: 'C:\ispTOOLS5_0_STRT\ispcpld\bin\vlog2jhd.exe "comparitor0.v" -p "c:\ispTOOLS5_0_STRT\ispcp


Done: completed successfully.
```

Figure 9.

Side Note (Controlling the pin assignments):

To have control over the pin assignments, you must declare the pin assignments in comments after you have declared the input or output variable. The format for a pin assignment declaration that is an input goes as follows (note: the X in PX is the pin number of the chip):

```
input  <variable name> /* synthesis LOC= “PX” */
```

and likewise the format for a pin assignment declaration that is an output goes as follows:

```
output <variable> /* synthesis LOC=“PX” */
```

The format for multiple pin assignments is as follows (note: once again X and Z are just pin numbers they can be any number you want):

input [n:0] <variable name> /* synthesis LOC= “P_n X….P_0Z” */

and likewise the format for a pin assignment declaration that is an output goes as follows:

output [n:0] <variable name> /* synthesis LOC= “P_n X….P_0Z” */

Here are some more examples of pin assignments:

Ex. 1 (input variable “flag” at pin 3):

```
input flag /* synthesis LOC=“P3” */;
```

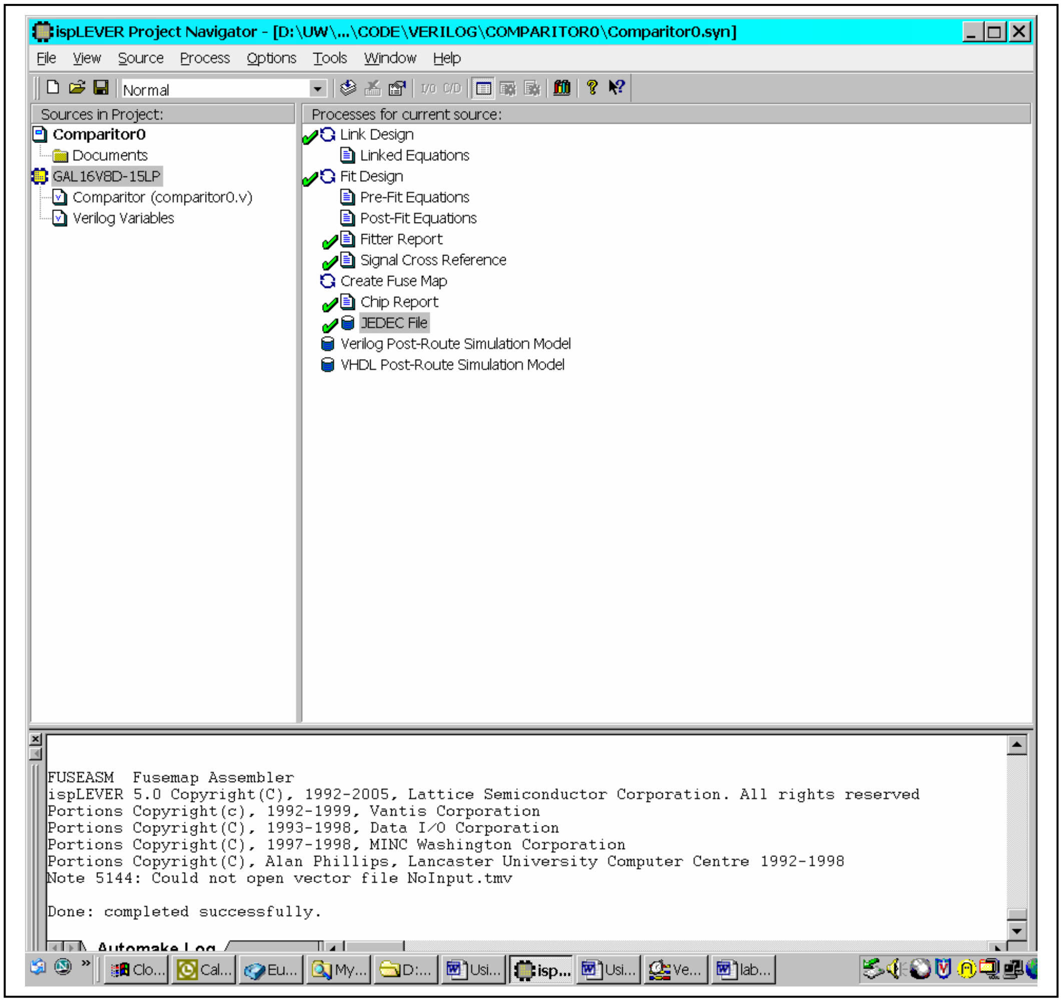
EX. 2 (Two output variables: carrryOut at pin 5 and barrowOut at pin 6):

```
output carryOut  /* synthesis LOC=“P17” */, borrowOut /* synthesis LOC=“P18”*/;
```

Note: if you do not put a space between the variable and the comment, Synplify (the compiler) will give you a warning.

13. To produce the JEDEC file, select the target chip in the *Sources in Project* pane and then double-click “**JEDEC File**” in the *Process for Current Sources* pane.

    Your *Process for Current Sources* pane should now look like that in Figure 10. The JEDEC file has been created and you’re ready to take it to the programmer! You’re finished!

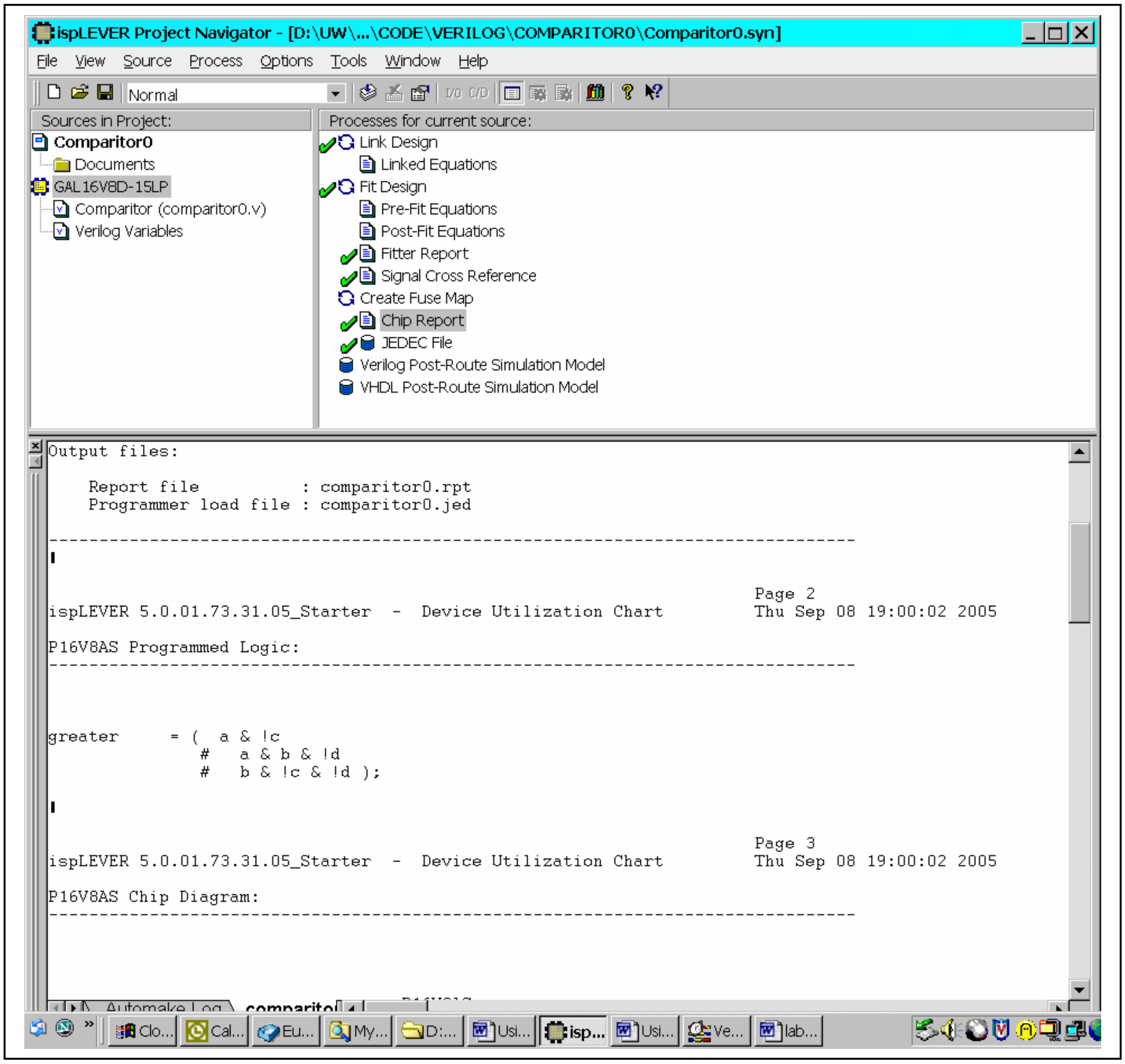

ispLEVER Project Navigator - [D:\UW\...\CODE\VERILOG\COMPARITOR0\Comparitor0.syn]
File View Source Process Options Tools Window Help
Normal
Sources in Project:
Comparitor0
Documents
GAL16V8D-15LP
Comparitor (comparitor0.v)
Verilog Variables
Processes for current source:
Link Design
Linked Equations
Fit Design
Pre-Fit Equations
Post-Fit Equations
Fitter Report
Signal Cross Reference
Create Fuse Map
Chip Report
JEDEC File
Verilog Post-Route Simulation Model
VHDL Post-Route Simulation Model
Automake Log
Clo... Cal... Eu... My... D:... Usi... isp... Usi... Ve... lab...

```
FUSEASM  Fusemap Assembler
ispLEVER 5.0 Copyright(C), 1992-2005, Lattice Semiconductor Corporation. All rights reserved
Portions Copyright(c), 1992-1999, Vantis Corporation
Portions Copyright(C), 1993-1998, Data I/O Corporation
Portions Copyright(C), 1997-1998, MINC Washington Corporation
Portions Copyright(C), Alan Phillips, Lancaster University Computer Centre 1992-1998
Note 5144: Could not open vector file NoInput.tmv

Done: completed successfully.
```

Figure 4

Chip Report – very handy!

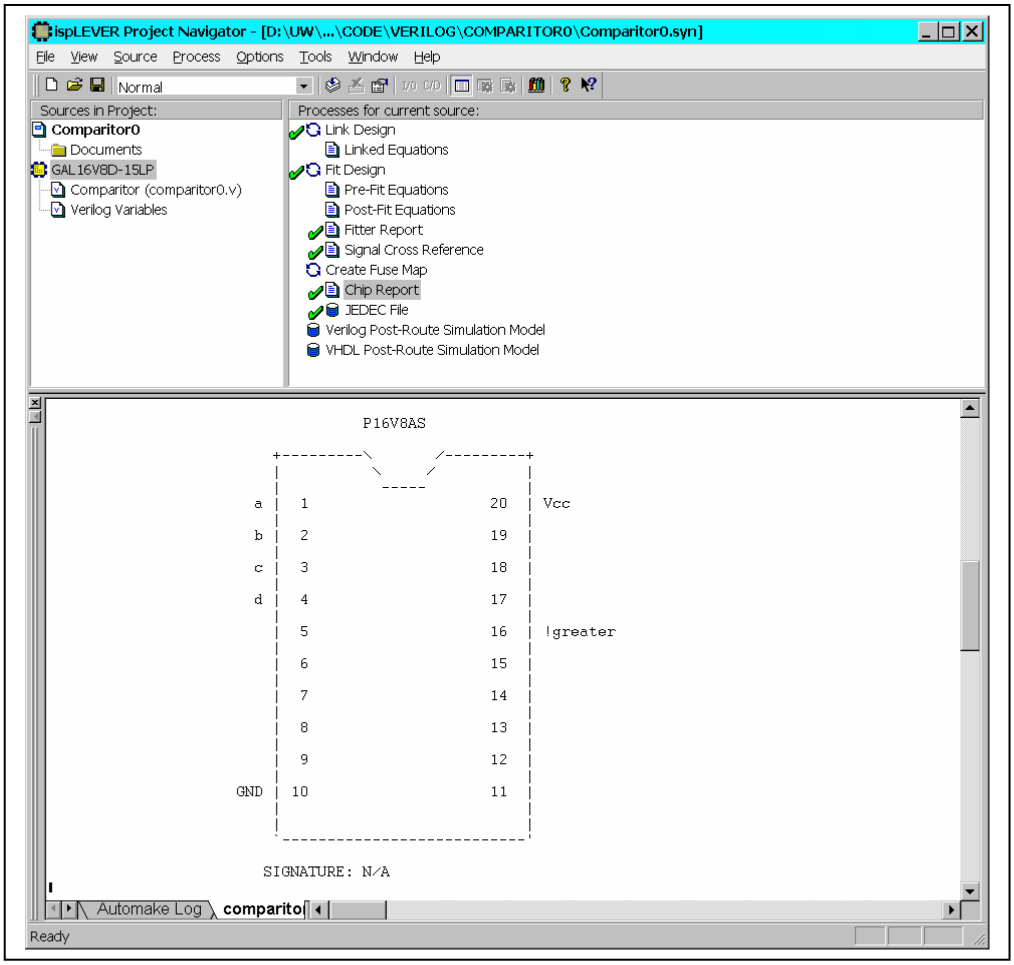
1. Double-click Chip Report in the Processes for Current Source window.
2. Open the Report Viewer window (which should now be displaying the Chip Report).
3. Scroll down and you'll see the equations that were created for your design as in Figure 11



```
Output files:

    Report file           : comparitor0.rpt
    Programmer load file  : comparitor0.jed

--------------------------------------------------------------------------------


                                                                Page 2
ispLEVER 5.0.01.73.31.05_Starter  -  Device Utilization Chart         Thu Sep 08 19:00:02 2005

P16V8AS Programmed Logic:
--------------------------------------------------------------------------------



greater     = (  a & !c
               #   a & b & !d
               #   b & !c & !d );



                                                                Page 3
ispLEVER 5.0.01.73.31.05_Starter  -  Device Utilization Chart         Thu Sep 08 19:00:02 2005

P16V8AS Chip Diagram:
--------------------------------------------------------------------------------
```

Figure 11

Scroll down further and you'll see a visual layout of the pin assignments (Figure 12)….which is very handy for wiring!
3.



```
                         P16V8AS

             +---------\       /---------+
             |          \     /          |
             |           -----           |
       a     |  1                    20  |   Vcc
             |                           |
       b     |  2                    19  |
             |                           |
       c     |  3                    18  |
             |                           |
       d     |  4                    17  |
             |                           |
             |  5                    16  |   !greater
             |                           |
             |  6                    15  |
             |                           |
             |  7                    14  |
             |                           |
             |  8                    13  |
             |                           |
             |  9                    12  |
             |                           |
     GND     | 10                    11  |
             |                           |
             `---------------------------'

          SIGNATURE: N/A
```

Figure 12