Use the WebPACK Project Navigator to Set Design Constraints in CoolRunner CPLDs

Here's an introduction to User Constraint Files for designing CoolRunner CPLDs with WebPACK, the complete design tool that you can download from the Web.

by Jennifer Jenkins
Applications Engineer, Xilinx Inc.
jennifer.jenkins@xilinx.com

WebPACK™ Project Navigator™ software allows you to control a variety of design attributes in CoolRunner™ XPLA3 CPLD designs. You can generally set a design constraint on a whole device through the WebPACK Navigator GUI. But if you want to set characteristics for an individual signal in your design, you have to set up UCFs. Setting user-defined constraint files is easy to do – and this allows you to maintain your design specifics throughout the iterations of the design life cycle.

Available Design Constraints

WebPACK Project Navigator gives you the capability to manually control the following attributes in a CoolRunner CPLD design:

- Pin and node assignments
- Output slew rate control
- Initial register state
- Input pin characteristics
- Signal optimization options.

The WebPACK Project Navigator GUI allows you to define some of these constraints for the device generally. Only a UCF, however, lets you set specific characteristics for an individual signal in a design.

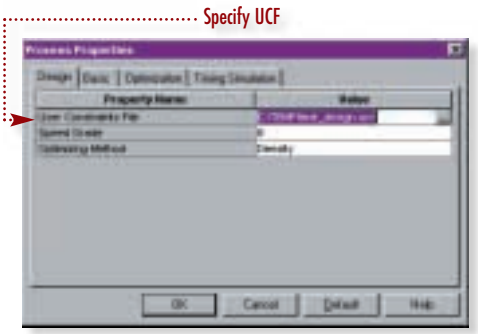

Figure 1 - WebPACK specification of UCF

Specification of Design Constraints

The WebPACK GUI allows you to specify a UCF during design implementation. The WebPACK Process Properties window in Figure 1 illustrates how to specify the UCF.

The WebPACK GUI can also automatically create a UCF that can assign pin locations. You can then edit the UCF to fit your design specifications. For instance, the following example shows how to assign a certain signal to a specific pin. Note the syntax depends on the type of package used.

For PLCC (Plastic Leadless Chip Carrier), PQFP (Plastic Quad Flat Pack), TQFP (Thin Quad Flat Pack), and similar packages, the syntax would be:

```
PIN <signal_name> LOC=Pnn;
```

and the code might read:

```
PIN qout<0> LOC = P20;
```

For CS (Chip Scale), BG (Ball Grid), and similar packages, the syntax would be:

```
PIN <signal_name> LOC=RC;
```

and the code might read:

```
PIN qout<1> LOC = G2;
```

Another attribute the UCF offers you is the capability to control the slew rate for a specific output signal. The WebPACK GUI allows you to set the slew rate for all outputs in a design. To control the slew rate for an individual output, the UCF syntax would be:

```
PIN <signal_name> <slewrate>;
```

and the code might read:

```
PIN qout<0> SLOW;
PIN qout<1> FAST;
```

These are just a few examples of the value UCFs bring to you. With UCFs, you can maintain consistency during design iterations and control a design during the implementation process.

For more information on using a UCF to enter design attributes, refer to XAPP352 "Utilizing a User Constraint File for CoolRunner CPLDs" (www.xilinx.com/xapp/xapp352.pdf).

To learn about the entire line of free, downloadable WebPACK ISE™ (Integrated Synthesis Environment) software, go to www.xilinx.com/webpack/index.html.