

Hierarchical Design Methodology with the Quartus Software

Technical Brief 59

October 1999, ver. 1



QUARTUS™

Altera Corporation
101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
<http://www.altera.com>
<https://websupport.altera.com>

High-density programmable logic devices (PLDs), such as APEX™ 20K devices, have created a shift in design methodology. To take advantage of high-density PLD capabilities, designers are moving away from schematic-based design techniques and moving toward hardware description languages (HDLs). As HDL designs grow more complex, more designers are shifting to a hierarchical design methodology for their digital designs.

The Quartus™ development tool is the first PLD software that optimizes the process of hierarchical design methodologies. This technical brief discusses the advantages of using hierarchical design techniques and explains how the Quartus software makes it easier to create hierarchical designs.

Hierarchical Design Benefits

The hierarchical design methodology involves dividing projects into an extensive tree of smaller design entities. Each entity has a specific function and a carefully declared method for interacting with other entities. In general, a hierarchical design methodology provides the following benefits:

- Allows designers to develop an organized design flow
- Allows designers to focus on the functionality of a single design entity
- Permits a high-level view of the design's overall structure, making it easier to uncover critical paths and potential trouble spots

Two fundamental practices in hierarchical design include the top-down and bottom-up design techniques. Both design techniques have their own strengths and allow designers to break large designs into smaller, more manageable pieces.

Top-Down Designs

In top-down methodologies, designers start with a high-level understanding of a design's overall flow. The general functionality of each design block is developed prior to creating the building blocks themselves. Top-down design techniques tend to create more organized designs because designers must first evaluate the design's overall structure. In addition, by focusing on structure and leaving details of a design undefined, designers can delay technical considerations until they are at more manageable levels.

Bottom-Up Designs

In bottom-up methodologies, designers first create the lower-level blocks and then integrate these blocks into higher-level structures. Designers focus their attention on individual sub-blocks rather than the design as a whole. Thus, individual blocks can be verified prior to building the entire design. Bottom-up designs also highlight the benefits of reusable code. Splitting projects into smaller modules helps designers find redundant sections in a project. Designers can then reuse individual blocks of code, which saves time. In addition, designers can assemble commonly used functions into their regular design library, allowing blocks of code to be reused from project to project.

The Quartus Software Simplifies Hierarchical Design

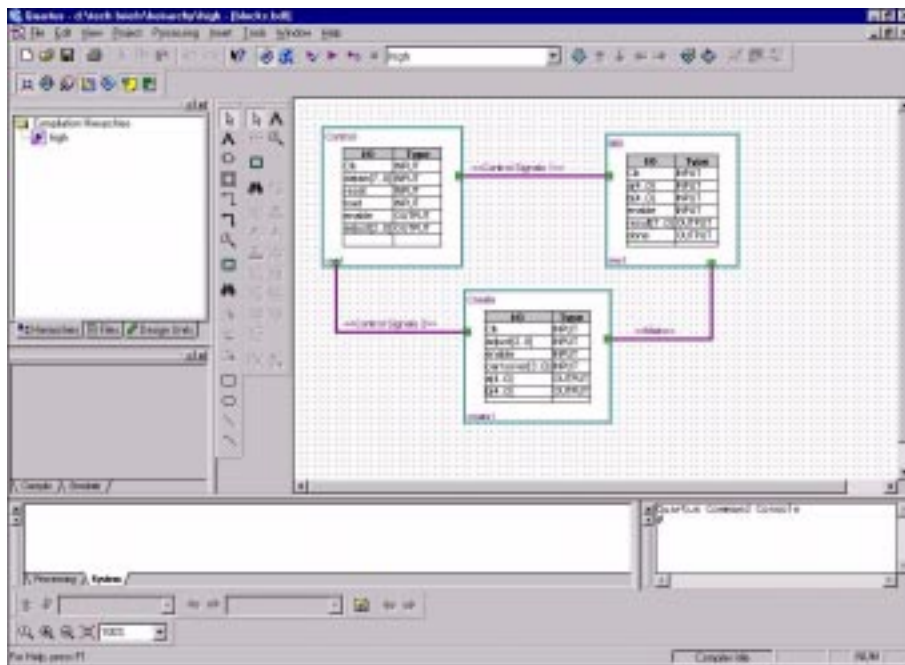
The Quartus software allows designers to use a top-down, bottom-up, or combined top-down/bottom-up design technique. The unique ability to use a hybrid approach offers designers the benefits of both design methods. For example, designers can still reuse individual blocks of code with the bottom-up approach, while evaluating the design’s overall structure before HDL coding begins with the top-down approach.

Bottom-up designs work the same in the Quartus software as they do in the MAX+PLUS® II software; lower-level design files are instantiated in higher-level design files through coding methods or through a schematic editor. Top-down designs, however, take advantage of two new Quartus software features. Designers can use the Quartus Block Editor and Smart Mapper to create a block-based graphical representation of top-down designs. Digital designs are constructed of “empty boxes”—graphical pieces representing a block of undefined code—that are connected together in a schematic format. Through these two features, designers can add, delete, and rearrange the design pieces while forming their overall design.

Block Editor

In the Quartus Block Editor, designers can use the Block Tool to create empty boxes for each functional piece of the design. Designers can define each top-level block’s properties, such as block name and I/O properties, using the **Block Properties** dialog box (Edit menu). [Figure 1](#) shows a sample top-level design file in the Quartus software.

Figure 1. Creating a Top-Level Design File in the Block Editor



After all top-level entities have defined properties, designers can use the Quartus software to create their HDL-based or schematic-based design files. The Quartus software generates an empty design file with the I/O ports already declared. For example, a VHDL-generated file for the MIX design entity in [Figure 1](#) would look like the file in [Figure 2](#).

Figure 2. Sample VHDL Design File

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

-- Entity Declaration

ENTITY mix IS
  -- {(ALTERA_IO_BEGIN)} DO NOT REMOVE THIS LINE!
  PORT
  (
    CLK      : IN STD_LOGIC;
    a        : IN STD_LOGIC_VECTOR(4 DOWNTO 0);
    b        : IN STD_LOGIC_VECTOR(4 DOWNTO 0);
    enable   : IN STD_LOGIC;
    result   : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
    done     : OUT STD_LOGIC
  );
  -- {(ALTERA_TO_END)} DO NOT REMOVE THIS LINE!

END mix;

-- Architecture Body

ARCHITECTURE MIX_architecture OF mix IS

BEGIN
END MIX_architecture;

```

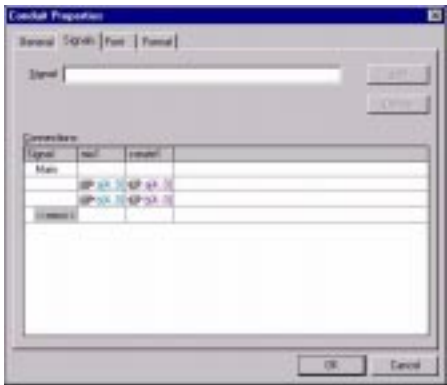
When the Quartus software generates a Block Editor design file, the designer can continue to use a top-down design method without developing any code. When designers reach the bottom-level design file, they only need to develop the design logic because the file's higher-level instantiations have already been defined by the Quartus software.

Smart Mapper

The Quartus software provides a simple way to connect top-level design entities. Rather than making connections for every I/O signal, the Quartus Smart Mapper allows designers to connect design entities through a conduit. Conduits graphically represent a collection of signals carrying information to and from design entities. A signal passes from one design entity to another through an unspecified bus, node, or signal in the conduit.

The nodes and buses included in each conduit are displayed under the **Signals** tab in the **Conduit Properties** dialog box. [Figure 3](#) shows the Main conduit information for the design in [Figure 1](#).

Figure 3. Information for the Main Conduit



Within a conduit, the Smart Mapper automatically connects two signals that appear in separate design entities and have the same name. Consequently, designers no longer need to manually connect signals if they share the same name. In addition, after a conduit has been created, a designer can add a signal to the design entity at any time, and the Smart Mapper will automatically add the signal to the conduit.

Summary

Designing in a hierarchical format is becoming a necessity as PLD designs become larger and more complex. Hierarchical design techniques help designers better organize and manage their designs by breaking them into smaller design entities. Having design software that easily supports both top-down and bottom-up design methodologies is key for today's designer. The Quartus software gives designers the tools necessary to take advantage of hierarchical design practices.

