

## Upgrading a DDR SDRAM Controller MegaCore Function v2.1.\* Design to v2.2.0

This white paper is intended for designers who have an existing FPGA design implementing the Altera® DDR SDRAM Controller MegaCore® Function version 2.1.0, 2.1.1, or 2.1.1 SP1 design (from now on referred to as 2.1.\*) and will guide you step-by-step through the upgrade process.

This white paper includes the following sections:

- Before you Begin
- Launch the Quartus II Software version 4.1
- Edit the MegaCore Function
- Design Updates
- Summary

### Before you Begin

Before you begin upgrading your design, follow these steps:

1. Obtain and install the Quartus® II software version 4.1, or higher.
  - You must use the Quartus II software version 4.1 or higher with the DDR SDRAM Controller MegaCore function v2.2.0. Timing assignments and location constraints in DDR SDRAM Controller v2.2.0 require Quartus II version 4.1 synthesis names.
2. Install the DDR SDRAM Controller MegaCore Function v2.2.0.
  - Altera recommends that you uninstall versions 2.1.\*, to ensure that all links to older versions are removed.
3. Edit the first line of your DDR SDRAM controller variation.
  - You must edit the existing DDR SDRAM v2.1.\* controller variation (**.v** or **.vhd**) file, so that the MegaWizard® Plug-In Manager sees the design file as if it were a v2.2.0 instance.
  - Open your v2.1.\* DDR SDRAM Controller variation file with a text editor. Edit the following line:
 

```
-- megafunction wizard: %DDR SDRAM Controller v2.1.*%
```

to

```
-- megafunction wizard: %DDR SDRAM Controller v2.2.0%
```

Save these changes.
4. Backup your current top-level design file (**.v** or **.vhd**).
  - When you click **Generate** in IP Toolbench, it will create a new top-level design file that has the same name as your top-level design name. Renaming the current top-level design file ensures it will not be overwritten.
  - However if IP Toolbench finds a top-level design file not created by version 2.2.0, it will save the old design file as *<filename>.old*.

### Launch the Quartus II Software version 4.1

Before you edit the existing MegaCore function, follow these steps:

1. Start the Quartus II software version 4.1 or higher.

2. Remove the v2.1.\* support files.
  - Choose **Open** (File menu) and choose your project **.qpf** file.
  - Choose **Settings > Files** (Assignments menu).
  - Choose all the files that begin with `auk_ddr` and click **Remove**.
3. Use the Quartus II Assignment Editor to edit your **.qsf** file.
  - You must remove all the old DDR SDRAM constraints, because the Quartus II software version 4.1 uses different synthesized net names.
  - Print out the v2.1.\* IP Toolbench **add\_constraints.tcl** file as a reference to see which assignments were made.
  - Choose **Assignment Editor** (Assignments menu) and choose **All** in the **Category** box, to make sure all of the current assignments are displayed.
  - Remove all of the old DDR SDRAM constraints, using **add\_constraints.tcl** as a reference.

## Edit the MegaCore Function

To edit the MegaCore function, follow these steps:

1. Edit your custom variation.
  - Choose **MegaWizard Plug-In Manager** (Tools menu) and select **Edit existing custom megafunction variation**.
2. Describe your hierarchy in the wizard.
  - The new Hierarchy tab allows you to enter the hierarchy path to the DDR SDRAM Controller in your design.
  - The hierarchy entered in the wizard must match your design, because, the constraints and timing scripts rely on this path for correct operation.
3. Generate your new v2.2.0 variation.
  - In IP Toolbench, click **Generate**.
  - IP Toolbench generates the following files in the language that you specified:
    - A new variation file
    - A new example top level design file
    - A clear text data path
    - A new testbench
    - A new **add\_constraints\_for\_<variation name>.tcl**
    - A new **verify\_ddr\_timing\_for\_<variation name>.tcl**
  - Also, IP Toolbench adds the new user library and support files to your project as required.
4. Examine the new example top-level design file.
  - The wizard generates a example top-level reference design in the language that you specified.
  - Keep this file as a reference.

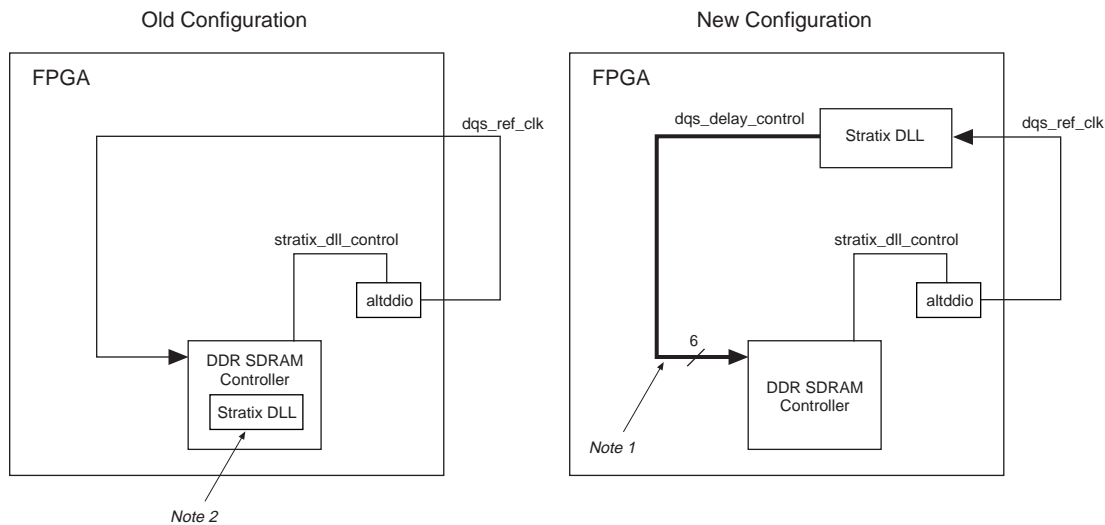
## Design Updates

To implement the design updates, follow these steps:

1. Change the byte order (v2.1.0 designs only)
  - The byte enable ordering was incorrect in v2.1.0, (fixed from v2.1.1 onwards); for v2.1.0 designs, check the order used.

2. Rename the new wizard-generated example top-level design file and re-introduce your original top-level design file.
3. Update the Stratix® DLL
  - Stratix brand products only, Cyclone™ users can skip to step 4.
  - The changes in Stratix brand product designs are the result of a simpler multiple core flow. The DLL is no longer instantiated inside the MegaCore function, which means that `dqs_ref_clk` is no longer connected to the MegaCore function. The `dqs_ref_clk` signal instead must be connected directly to the DLL. Figure 1 shows the old and new configurations.
  - The MegaCore function now uses the DLL output (the `dqs_delay_control` signal) as an input to control the settings for the DQS pins. The `dqs_delay_control` signal tells the DQS pins how much delay to apply to the incoming DQS signal.
  - Consult the example top-level design file for your exact configuration in your specified HDL language.
  - You can copy and paste the Stratix DLL instantiation directly into your top-level design file.

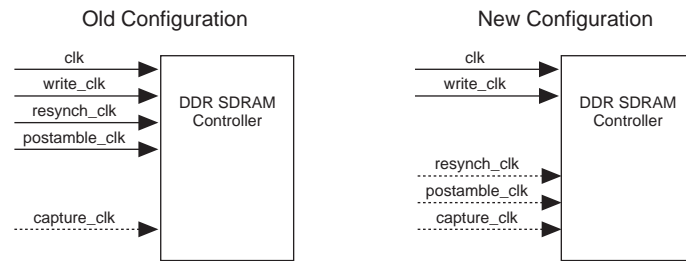
Figure 1. Stratix DLL

**Notes:**

- (1) `dqs_delay_control` is a six-bit bus for Stratix II devices. In Stratix devices four of the bits are unused.
- (2) In the old configuration the Stratix DLL is inside the DDR SDRAM controller

4. Configure the clocks in your design.
  - The connection of the postamble and resynchronization clocks in version 2.2.0 has been simplified (see Figure 2). In most cases these connections are made inside the DDR SDRAM controller.
  - There may no longer be a `resynch_clk` or `postamble_clk` pin on your variation, unless a dedicated clock is required for these clocks.

Figure 2. Clock Configuration (1)

**Note:**

(1) The dotted lines represent optional clocks. Any of these clock may be required, depending on your system configuration.

## 5. Run the new constraints script for the design.

- If you are using the recommended pin names, e.g., `<prefix>_dqs`, you can run the constraints script without modifying it. If not follow these steps to make your script match your system:
  - Open the file `add_constraints_for_<variation name>.tcl` with a text editor.
  - Scroll down to line 36.
  - You can change the pin names to be the actual pin names used in your design, if they are different.
  - Check the hierarchy and clock pin names, adjust these to match your design if necessary.
  - Save any changes.
- Choose **Tcl Script** (Tools menu).
- Choose `add_constraints_for_<variation name>.tcl` and click **Run**.

## 6. Check the constraints have been applied.

- Choose **Start > Start Analysis & Synthesis** (Processing menu).
- Open the Quartus II Assignment Editor and check that the assignments have all been applied.
- Investigate any "?"s. "stratix\_dqs\_delay\_buffer" can be ignored if you have not selected additional DQS delay matching buffers in IP Toolbench. All other "?"s may point to issues with the hierarchy or pin names and must be resolved to stop the Quartus II software from optimizing parts of the design away.

## 7. Compile design.

- Move all of the support files to the top of the list (i.e., the bottom of the hierarchy).
  - Choose **Settings** (Assignments menu).
  - Click **File** in the Category list.
  - Choose all files that begin with `auk_ddr` and click **Up** enough times to move the files to the top of the list.
- Click **Compile** (Processing menu).
- If you receive the following error, ensure you have moved the all of support files to the top of the file list:  
 Error: VHDL use clause error at <design>: design library `auk_dd_use_lib` does not contain primary unit `auk_ddr_tb_function`

## 8. Run the new verify DDR SDRAM timing flow.

- Choose **Run Tcl Script** (Tools menu).
- Choose `verify_ddr_timing_for_<variation name>.tcl` and click **Run**
- The Tcl script extracts and analyses DDR SDRAM specific timing nodes from the Quartus II database.
- The script reports an error if certain nodes have not been correctly applied or if part of the local read data bus is not connected in your design.

## Summary

Use the check list (see Table 1) to ensure that you have performed all of the necessary steps to upgrade your DDR SDRAM Controller MegaCore function version 2.1.\* to version 2.2.0.

Table 1. Check List

Operation	Check
Obtain and install the Quartus® II software version 4.1, or higher.	
Install the DDR SDRAM Controller MegaCore Function v2.2.0.	
Edit the first line of your DDR SDRAM controller variation.	
Backup your current top-level design file (.v or .vhd).	
Start the Quartus II software version 4.1 or higher.	
Remove the v2.1.* support files.	
Use the Quartus II Assignment Editor to edit your .qsf file.	
Edit your custom variation.	
Describe your hierarchy in the wizard.	
Generate your new v2.2.0 variation.	
Examine the new example top-level design file.	
Change the byte order (v2.1.0 designs only)	
Rename the new wizard-generated example top-level design file and re-introduce your original top-level design file.	
Update the Stratix® DLL	
Configure the clocks in your design.	
Run the new constraints script for the design.	
Check the constraints have been applied.	
Compile design.	
Run the new verify DDR SDRAM timing flow.	



101 Innovation Drive  
San Jose, CA 95134  
(408) 544-7000  
www.altera.com

Copyright © 2004 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.