White Paper

DDR & DDR2 SDRAM Controller Compiler FAQ

Introduction

The Altera® DDR & DDR2 SDRAM Controller Compiler frequently asked questions (FAQ) white paper discusses the following topics:

For information on efficiency and latency, refer to the *Improving the Efficiency of the DDR & DDR2 SDRAM Controller Compiler White Paper.*

Hardware

Why does IP Toolbench sometimes choose the "72° " not the "90° " phase shift for the DQS?

DQ and DQS leave the memory device edge aligned. The DQS must be delayed inside the FPGA to place the strobe in a good position with respect to its associated data group. Some delay occurs outside of the FPGA. The process, voltage, and temperature (PVT) variations across the Altera device, the memory device and the board must be taken into consideration.

When these variations are considered and put over the top of each other a "window" of good sample time is left. The scripts within IP Toolbench pick a time close to the middle of this window (to give maximum margin). Frequently, this action results in a requirement for a phase shift of 72° within the FPGA section of the DQS path of the whole system.

Why is the DQS phase reference circuit turned on and off?

Allowing the Stratix® DLL to free run is a perfectly valid mode of operation. However, any jitter can potentially manifest itself as a reduction in f_{MAX}.

To combat any jitter effects ensure the DLL is not updated during reads. A convenient point at which to update the DLL is the refresh period. The DLL selects the appropriate delay chain, for the given frequency and temperature. When the DLL is not receiving the clock, its output is stable and fixed, and there are no jitter effects.

Can I use a different input standard for the PLL input?

Stratix PLLs have their own isolated input banks, which means that they can have a different input standard compared to the rest of the bank. However Cyclone™ devices need defined a 2.5-V input for the PLL, because it is in the same bank and different I/O standards can not be placed in the same bank.

See the PLL configuration section in the *DDR & DDR2 SDRAM Controller Compiler User Guide*, for further details of the circuit to implement.

Why does the circuit not operate when the clock frequency is reduced?

For each setting of the MegaCore function, there is an upper and lower frequency limit. After targeting and compiling the MegaCore function for 100 MHz, do not turn the clock down and assume it will run at 80 MHz. The IP Toolbench timing analysis looks at whether the target frequency can be achieved and sets up the core assuming that frequency is in use. For example, the core uses resynchronization registers. IP Toolbench looks at the timing information that you input, and chooses a clock edge that is appropriate for that resynchronization. Subsequently changing the frequency means that the data may no longer be synchronized correctly, hence the system fails. (See also “Why does IP Toolbench rather than the Quartus II software perform the timing analysis?” on page 2).

Are there any board guidelines?

The following sources give information on board guidelines:

- *Application Note 348: Interfacing DDR SDRAM with Cyclone devices*
- Board Guidelines Solution Centre on the Altera web site, **www.altera.com/support/devices/board/brd-index.html**
- The vendor’s web site of your chosen DDR or DDR2 SDRAM

Note: to predict the best termination arrangements and board layout, you should also simulate your design including the PCB and the device package.

Timing

Why does my Cyclone board run in excess of 133 MHz but Altera only guarantees 133 MHz?

It is recognized that any particular board can be “tuned” to achieve frequencies well above those that Altera guarantees. However, any figures released by Altera must be achievable across the entire process, voltage, and temperature range for the FPGA and a 133 MHz DDR SDRAM memory device in the worst conditions and allow some margin for reasonable variation in layout.

Why does IP Toolbench rather than the Quartus II software perform the timing analysis?

The Quartus® II software is a device-level tool. To correctly analyze the complete round trip timings for the DDR or DDR2 SDRAM Controller requires a higher system-level perspective. The IP Toolbench script uses the following information:

- Quartus II device timing analysis results

- Data from the memory part specifications
- Data about board delays and skews.

Ensure that all of the above data is input correctly into the wizard, otherwise, the core may fail to operate as desired. See also “Why does the circuit not operate when the clock frequency is reduced?” on page 2.

What happens if the timing analysis suggests my configuration will fail timing by 5 ps?

It may be that another clock edge (one not chosen automatically by the scripts) can provide an appropriate clock to allow the system to operate as desired. To adjust the resynchronization and/or postamble clock phase, follow these steps:

- Launch IP Toolbench.
- Open the Timing Estimates tab
- Click on the Advanced tab
- Set to manual and adjust the clock edge whilst monitoring the set-up and holds times. The objective is to get the best set up and hold balance.

Can I run at frequencies other than 100, 133, 167 or 200 MHz?

Yes, the frequency input box on the IP Toolbench is a “free” text format box. To run the core at a particular frequency, perhaps to achieve some specific bandwidth, set the core up for operation at the desired frequency. Note, also change the PLL to be set-up for this frequency.

What happens if the skew on my board is outside recommendations?

Return to the timing analysis and place in the real skew numbers. If the timing analysis suggests that the desired frequency of operation can still be met, continue.

What is fed-back clock?

Fed-back clock is a scheme that compensates for the PVT variation in the output buffer and trace length and increases the resynchronization margin. The phase uncertainty between fed-back clock and the internal FPGA memory clock is less than the phase uncertainty between the DQS and the FPGA memory clock, which increases the timing margins for data transfer to the FPGA memory clock.

Simulation

My simulation reports timing violations, what are the most likely causes?

Timing violations (the data and local-data signals are not aligned) are probably setup and hold violations that are being caused because there are no delays in the testbench that represent the delay between FPGA and DDR or DDR2 SDRAM. Although IP Toolbench asks for the board delay and skew this information determines the correct resynchronization clock phase, it is not incorporated in the model in any way (as the model only represents the logic being placed inside the FPGA). So this board level delay information also needs to be put into the testbench. If there are no delays in the testbench, there is just the usual problem of testing a gate-level module with an RTL testbench. Edges returning from the testbench to the gate-level module are occurring too close to the clock edge. Refer to the IP Toolbench-generated testbench for a reference point.

Quartus II Software Topics

Why does the verify timing script fail to run?

If any of the following scenarios are true the timing script fails to run and you may receive the following errors:

- Half (or more) of the local data bus is not connected—most likely the read data bus
- The controller requires a dedicated resynchronization clock and one has not been wired up

- Read request is tied low
- The add constraints script has not been applied

```
Error: Can’t find path to pin 'clk_to_sdram0' (timing node =
'clk_to_sdram0_out')

Error: The timing node name ‘ ….. |ddr_sdram_core_auk_ddr_sdram: ……
dq_captured_rising[0]~ddio1_' was expected to be found in the timing netlist
but could not be found

Warning: Atom ddr_auk_ddr_sdram:ddr_auk_ddr_sdram_inst ….
|dq_captured_rising[7] has port DDIOREGOUT that should be connected in DDIO
input and bidirectional modes
```

If any of the following scenarios are true the timing script fails to run and you may receive the following errors:

- The clock pin names specified in IP Toolbench do not match the clock names in your design. Note: IP Toolbench treats the clock names as a bus
- The add constraints script has not been applied

```
Error:  Timing node 'ddr_0_clk_p_out' (pin name = 'ddr_0_clk_p') cannot be
found

Extra Info:       |__  The possible cause of the above error is that you have
not entered the correct positive clock pin name for variation 'ddr_sdram_0'.“
```

The assignment editor has a lot of little “?”s, what are these?

If all of these location region assignments have a question mark, the hierarchy path to the controller may have been changed. If the instance name has changed or has been moved into another block of logic, perform a search and replace on the assignments. These assignments constrain the resynchronization registers and are critical.

Why do I get this error: Quartus Error: DQS I/O pin <path> must have a combinational output to the device?

Either the input or output section of the data path is not correctly or fully wired up. Commonly, it is the read section that is not wired up fully. For example, it may be that only the bottom 8 bits are wired up (perhaps so that a quick compile can be tried), which causes the error. To resolve the error, ensure that all the data path is wired (even if only through a giant `OR` gate with single output fed to a pin) to stop the Quartus II software optimizing the circuit away.

Miscellaneous Topics

Why is the maximum data width for Cyclone Devices restricted to 48 bits?

There are a total of 8 global clocks in a Cyclone device. 2 are used in the DDR or DDR 2 SDRAM Controller, which leaves a total of 6 global clocks. The Quartus II software automatically routes the DQS through a delay to get the 90° shift then places the DQS signal onto a global clock signal. Thus the maximum data width for a Cyclone device is 48 bits. i.e, 6 × 8 = 48, where 8 is the width of each DQ group and 6 is the number of “available” clocks.

How do I choose a refresh period?

The DDR or DDR2 SDRAM Controller completes the transaction it is involved in before it actions the refresh. Thus although the refresh period, shown in IP Toolbench, is loaded into a counter and is always a constant, the time at which the refresh occurs, can vary. For example, assume the DDR or DDR2 SDRAM Controller has just started a

burst of 8 as the counter times out. There will be a further 3 cycles before the refresh actually occurs. To compensate for this irregularity, deduct a few cycles from the refresh count (in IP Toolbench), so that on average, the refresh becomes the desired number of cycles. To be very conservative, subtract 8 cycles for the worst case.

For self-testing can I decouple the DDR or DDR2 SDRAM controller from the rest of the design?

For self-testing, you can use the IP Toolbench-generated example instance (**_top.vhd** or **.v**), which writes data, reads data back, and compares the two. It displays the result as pass not fail (pnf). This example instance is stand-alone and allows you to debug the DDR or DDR2 SDRAM interface without effects from the rest of the system.

Where can I get more information?

The Altera website at **www.altera.com/technology/memory/mem-index.jsp** is a single point for DDR and DDR2 SDRAM related resources and contains the latest information and the following resources:

- Reference designs
- White papers
- Test results
- Boards and kit information

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