User Guide to Determine the Optimal DCM Phase Shift for the DDR Feedback Clock

UG178 (v1.5) June 18, 2007

Revision History

The following table shows the revision history for this document.

	Version	Revision
07/8/05	1.0	Initial Xilinx release.
8/24/06	1.1	Updated for EDK 8.1i
9/29/06	1.2	Updated for EDK 8.2.01i
3/2/07	1.3	Updated for EDK 8.2.02i
4/24/07	1.4	Updated for EDK 9.1i
6/18/07	1.5	Updated for EDK 9.1.02ii

Table of Contents

About This Guide

This reference system allows the user to find the optimal DCM phase shift for the DDR feedback clock. The completed EDK project is on the reference CD in the `reference_systems/EDK_projects/ml403_dcm_phase_shift` directory.

This document describes the EDK project, the phase shift software application, and how to run the EDK project.

Guide Contents

This user guide contains one chapter:

- User Guide to Determine the Optimal DCM Phase Shift for the DDR Feedback Clock," page 9

Additional Resources

To find additional documentation, see the Xilinx website at:

http://www.xilinx.com/literature/index.htm.

To search the Answer Database of silicon, software, and IP questions and answers, or to create a technical support WebCase, see the Xilinx website at:

http://www.xilinx.com/support.

Conventions

This document uses the following conventions. An example illustrates each convention.

Typographical

The following typographical conventions are used in this document:

Convention	Meaning or Use	Example
`Courier font`	Messages, prompts, and program files that the system displays	`speed grade: - 100`
`Courier bold`	Literal commands that you enter in a syntactical statement	**`ngdbuild`** *design_name*

Convention	Meaning or Use	Example
Helvetica bold	Commands that you select from a menu	**File → Open**
	Keyboard shortcuts	**Ctrl+C**
Italic font	Variables in a syntax statement for which you must supply values	**`ngdbuild`** *design_name*
	References to other manuals	See the *Development System Reference Guide* for more information.
	Emphasis in text	If a wire is drawn so that it overlaps the pin of a symbol, the two nets are *not* connected.
Square brackets [ ]	An optional entry or parameter. However, in bus specifications, such as **`bus[7:0]`**, they are required.	**`ngdbuild`** `[`*`option_name`*`]` *`design_name`*
Braces { }	A list of items from which you must choose one or more	**`lowpwr`** **`={on|off}`**
Vertical bar |	Separates items in a list of choices	**`lowpwr`** **`={on|off}`**

Vertical ellipsis
.
.
.
	Repetitive material that has been omitted	`IOB #1: Name = QOUT’`
		`IOB #2: Name = CLKIN’`
		.
		.
		.

Horizontal ellipsis . . .	Repetitive material that has been omitted	**`allow block`** *block_name loc1 loc2 ... locn;*

Online Document

The following conventions are used in this document:

Convention	Meaning or Use	Example
Blue text	Cross-reference link to a location in the current document	See the section “Additional Resources” for details. Refer to “Title Formats” in Chapter 1 for details.
Red text	Cross-reference link to a location in another document	See Figure 2-5 in the *Virtex-II Handbook.*
Blue, underlined text	Hyperlink to a website (URL)	Go to http://www.xilinx.com for the latest speed files.

User Guide to Determine the Optimal DCM Phase Shift for the DDR Feedback Clock

Abstract

This reference system determines the optimal phase shift for a DDR memory feedback clock on the ML403 development board. This reference system uses the GPIO output to control the output phase shift of the DCM. The GPIO output is controlled by a software application that is run on a MicroBlaze™ microprocessor. This application is run out of an internal FPGA BRAM and it traverses the entire range of possible phase shifts (-255 to 255) that correspond to the complete 360 degree phase shift cycle. At each phase shift value, the application performs memory tests and records if the tests passed or failed. The passing ranges are reported by printing them to a hyperterminal through a UART. The optimal phase shift values are calculated by choosing the middle values for the passing ranges.

Included System

Included with this user guide is one reference system located on the reference CD at:

- reference_systems/EDK_projects/ml403_dcm_phase_shift

Note: The **sw** directory, which holds all of the source files for the software applications, must be two directory levels above the directory where the reference system project file resides. This **sw** directory holds all of the source files for the DCM Phase Shift reference system.

Introduction

When building a new circuit board that uses an FPGA and a DDR (Double Data Rate) memory, it is necessary to know the optimal phase shift for the DDR feedback clock. This is because double data rate memory controllers write/read data at both the falling and rising edges of the controller clock. The optimal time to sample the data is in the middle of each of the half clock cycles, or 90 and 270 degrees (See Figure 1). To ensure that the sampling occurs at the desired times, it is necessary to account for the routing delay of the DDR feedback clock. In a Xilinx FPGA system this can be accomplished with the help of a Digital Clock Manager (DCM). The DCMs that are available in most Xilinx FPGAs have a phase shifter component. So if the DDR feedback clock is fed into a DCM, the phase of the output clocks can be adjusted so that the 90 and 270 degree shifted clocks rise in the middle of the controller clock half periods (See Figure 1). The only difficult part with this approach is knowing exactly how much to shift the phase. This amount is dependent not only on how much routing delay there is from the DDR chip to the FPGA, but also how much delay there is from the FPGA input pin to the DCM. This reference system provides a simple system to find this optimal phase shift ($t_{CLK\_90\_PS}$) without having to know anything about the routing delays.

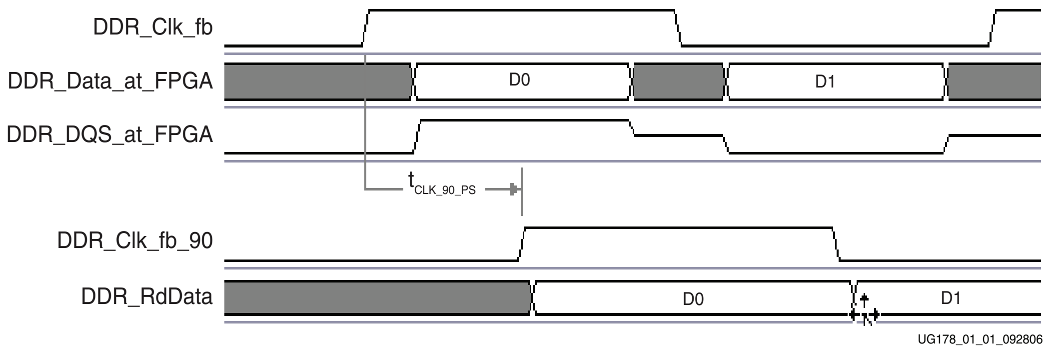

Figure 1: **DDR Data Capture**

Reference System Specifics

This reference system is built on the ML403 development board. The system uses the MicroBlaze (MB) microprocessor and the MCH OPB DDR memory controller. Figure 2 shows a high level block diagram of the system. The address map for this reference system is shown in Table 1.

The bitstreams for this system is available in **ready_for_download/** under the project root directory. The `download.bit` file downloads the system and the software application to the board.

Block Diagram

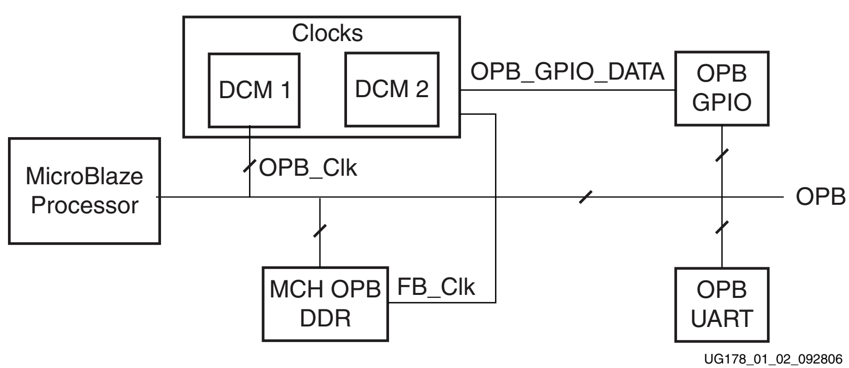

Figure 2: **Reference System Block Diagram**

Address Map

Table 1: **Address Map for Reference System**

Instance	Base Address	High Address
dlmb_cntlr	0x00000000	0x00003FFF
ilmb_cntlr	0x00000000	0x00003FFF
debug_module	0x4140000	0x4140FFFF
RS232_Uart	0x40400000	0x4040FFFF
DDR_SDRAM_64Mx32	0x24000000	0x27FFFFFF
opb_gpio_0	0x40000000	0x4000FFFF

What is Different from a Standard BSB Build

There are two changes that have been done in this reference system to use the phase shifting application. The clock is changed so that the phase shifting of the DCM that is fed by the DDR feedback clock is variable and the inputs are connected to the outputs of the GPIO core. The second modification is changing the sample application so that it performs the phase shifting by controlling the outputs of the GPIO.

Changing the Clocking Structure and Other Related Hardware

In the first step in modifying the clocking structure, the clocks module available in the pcores directory is added to the list of peripherals from the IP Catalog tab, as in Figure 3.

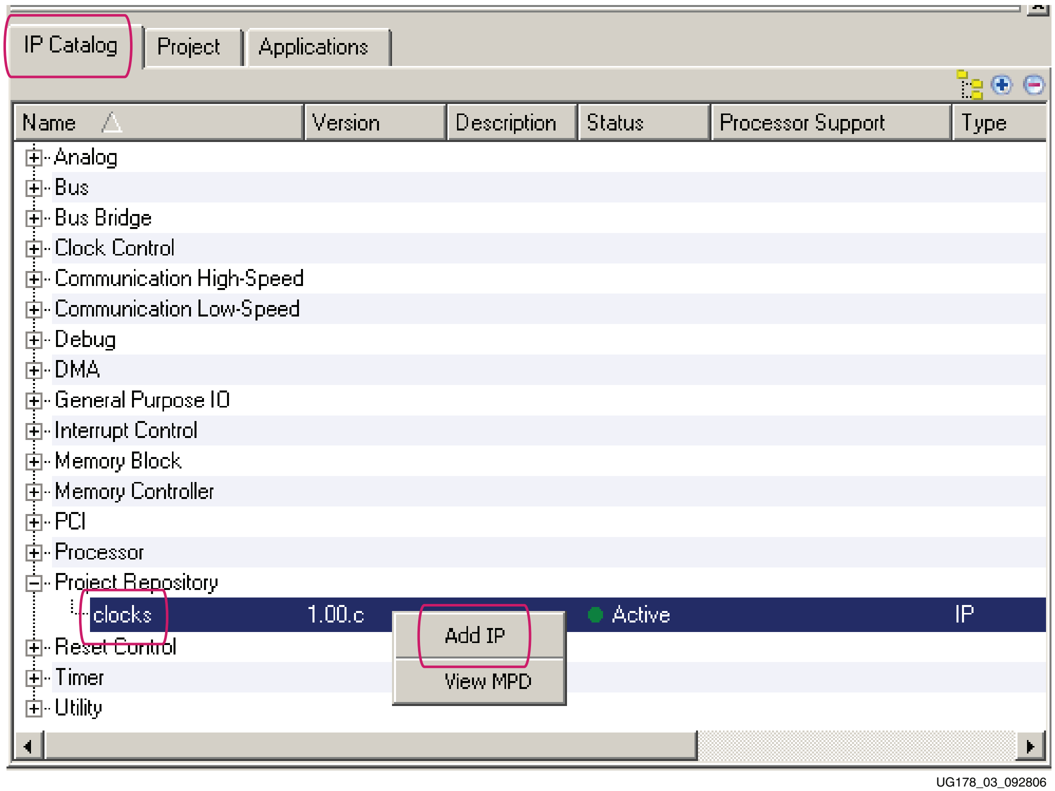

Figure 3: **Product Information Area / IP Catalog Tab, Add IP**

The ports of the clock module are added in the System Assembly View, Ports Filter selection, and hooked up to the following connections as described in Table 2 and shown in Figure 5. The block diagram of the clocks module connections are seen in Figure 4.

Table 2: **Clocks Module Ports and Connections**

Port Name	Connection	I/O	Description
dcm_clk	dcm_clk_s	I	This is the raw system clock and must be tied to the signal that is connected to the clk pin of the FPGA
dcm_rst_n	sys_rst_s	I	This is the system reset pin. It must be tied to the signal that attaches to the reset pin of the FPGA
sys_clk	sys_clk_s	O	This is the opb clock.
sys_clk_n	sys_clk_n_s	O	This is the 180 degrees out of phase bus clock
sys_clk90	clk_90_s	O	This is the 90 degrees out of phase bus clock
sys_clk270	clk_90_n_s	O	This is the 270 degrees out of phase bus clock
ddr_fb_clk	ddr_feedback_s	I	This is the DDR memory feedback clock
ddr_fb_clk90	ddr_clk_90_s	O	This is the DDR feedback clock 90 degrees out of phase. This clock is phase shifted so that it rises in the middle of the first half of the clock cycle

Table 2: **Clocks Module Ports and Connections**

Port Name	Connection	I/O	Description
ddr_fb_clk270	ddr_clk_90_n_s	O	This is the DDR feedback clock 270 degrees out of phase. This clock is phase shifted so that it rises in the middle of the second half of the clock cycle
dcm_locked	clocks_0_dcm_locked		Unconnected
ps_in	gpio_data_out[0]	I	This is the phase shifting input that is driven by one of the GPIO outputs to tell the DCM when to shift to the next phase. This input signal is tied to the signal that is connected to GPIO_d_out of the 2-bit wide GPIO core. Connect to bit 0
ps_incdec	gpio_data_out[1]	I	This controls the phase of the DCM outputs by incrementing or decrementing. This signal is tied to the other bit of the GPIO_d_out port of the 2-bit wide GPIO core. Connect to bit 1

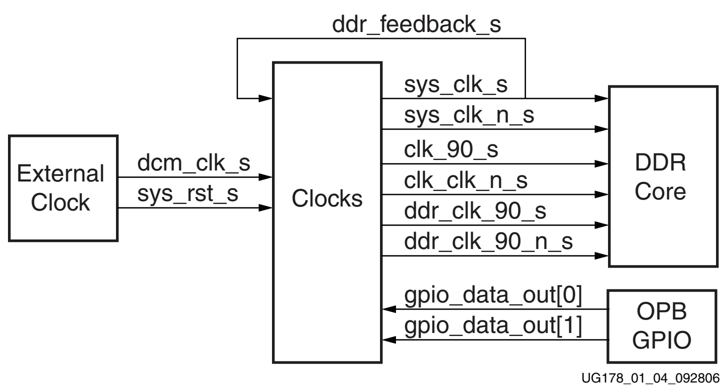

Figure 4: **Clocks Module Connections Block Diagram**

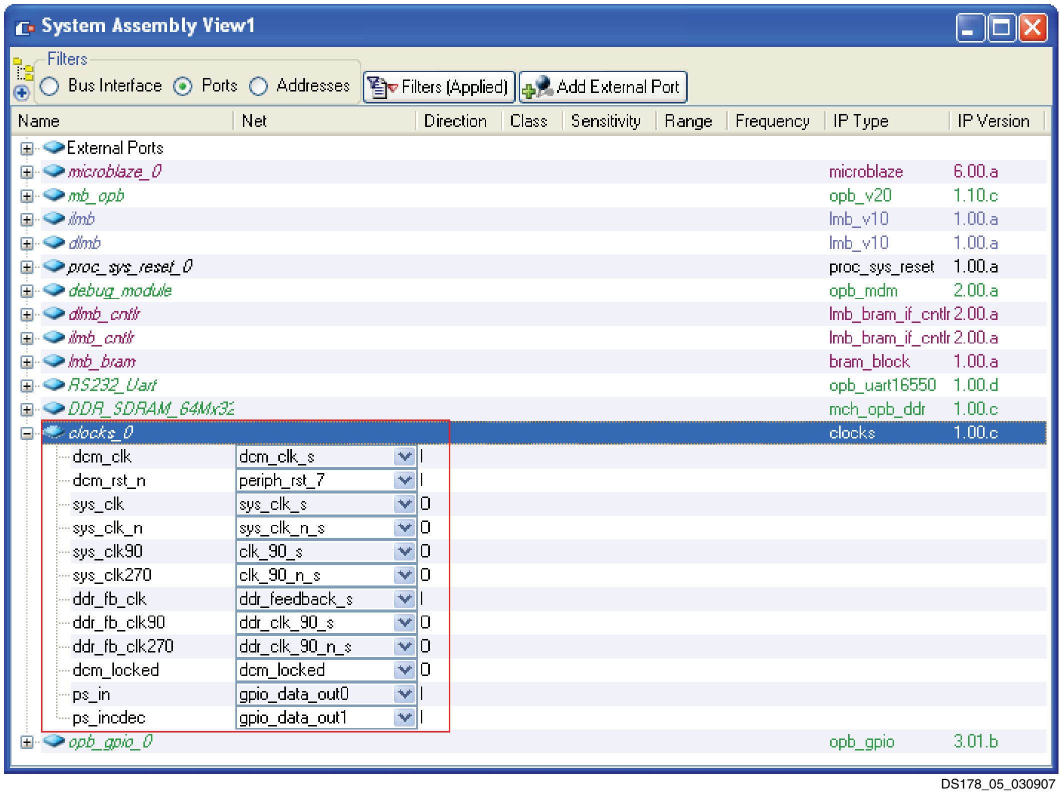

Figure 5: **System Assembly View / Ports Filter, Clock Module Connections**

The ps_in and ps_incdec ports from the clocks module are connected to gpio_data_out[0] and gpio_data_out[1], respectively. This 2-bit wide gpio_data_out signal is connected to the GPIO core as shown in Figure 6.

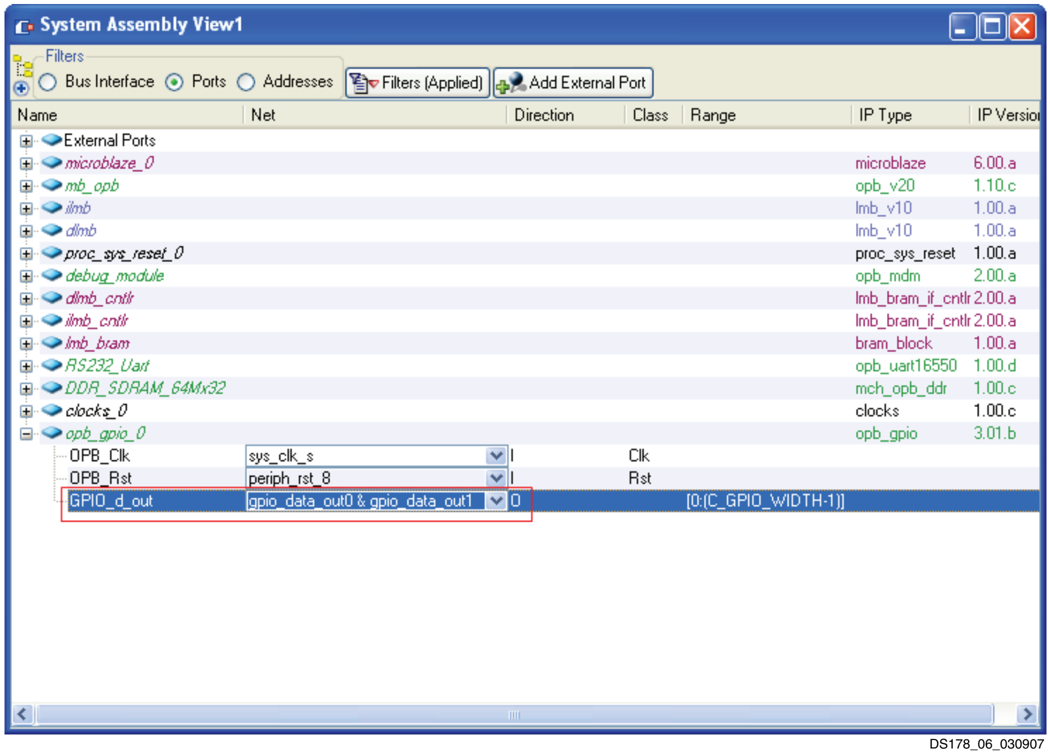

Figure 6: **System Assembly View / Ports Filter, OPB_GPIO Module Connections**

In the last modification, the DCM that created the bus clock (also the 90, 180 and 270 degrees out of phase bus clocks) is deleted from the design, since the DCMs are included in the clocks module.

The Phase Shift Software Application

The sample application that is initially created by BSB is replaced by the phase shift software application. It is compiled to run out of internal BRAM.

The phase shift software application starts with the phase of the DDR feedback DCM output phase set to 0. The phase is then decremented to -255 (the lowest possible phase shift), by pulsing the OPB GPIO output 255 times with the increment/decrement signal set to decrement. This information is passed to the DCM that actually shifts the phase. Once the phase is set to -255, the phase is incremented one by one to 255 (the highest possible phase shift). At each phase value, the application performs 32-bit, 16-bit and 8-bit memory tests and records if the tests passes or fails. Since the DDR memory controller writes/reads data during both the negative and positive portions of the clock cycle, we can expect there to be two passing ranges - one during the positive half and the other during the negative half.

Executing the Reference System

Generate the bitstream or use the provided files in the **ready_for_download** directory. Start a HyperTerminal session and connect the COM port to the UART on the board. Set the baud rate to 9600 and data bits to 8. See Figure 7 for the HyperTerminal settings. The UART terminal captures the results of the phase shift test.

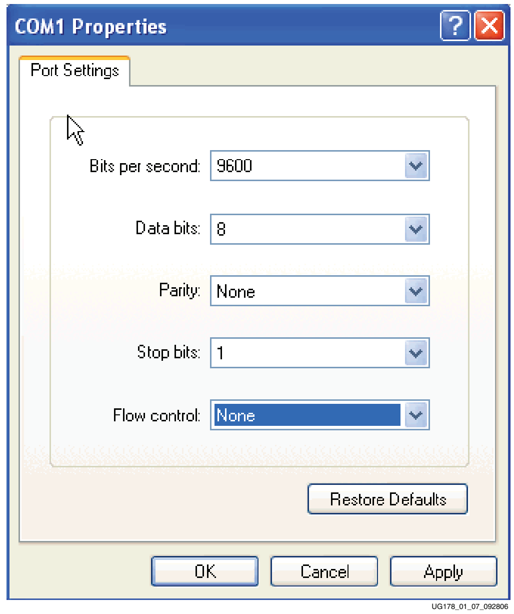

Figure 7: **HyperTerminal Settings**

Download the application to the board. The output should look similar to this after finishing downloading:

```
DCM PHASE SHIFT TEST

____________________


Moving the initial phase to -255

........................................................................

........................................................................

........................................................................

...............  Done


Incrementing through all the phases and reporting the passing ranges


Passing range is -21 to 86, optimal phase shift for this range is 32


Set your DDR feedback clock phase shift to 32


Finished testing
```

The positive optimal phase shift is the number that is used for the actual phase shift for the DDR feedback clock in the final system.

Modifying the DCM Phase Shift for the DDR Feedback Clock in the System

Once the phase shift application determines the optimal DCM phase shift for the DDR feedback clock on the board, it is necessary to change it in the final design and fix the phase shift so it is no longer variable. In the case of the output from the previous section, i.e. optimal phase shift is 32, the DCM parameters are as follows:

```
// synthesis attribute CLKOUT_PHASE_SHIFT    of dcm1 is "FIXED"

// synthesis attribute PHASE_SHIFT           of dcm1 is "32"
```

In theory, the design should also work with the phase shift of the DCM set to the optimal value from the negative range, but all that really does is set all the clocks 180 degrees out of phase. What used to be clk_90 in reality becomes clk_270, etc.

Note that the optimal phase shift value that the software application provides is specific to your board and also the DCM used. So if you create a new design and use a different DCM, the optimal phase shift value could also change. This is why it is necessary to constrain the DCM in each design. This is accomplished by editing the data/system.ucf constraints file. For instance, the following constraint locks DCM1 to a specific DCM in the FPGA.

```
inst "clocks_0/clocks_0/dcm1" LOC = DCM_X1Y1;
```

References

XAPP806 Determining the Optimal DCM Phase Shift for the DDR Feedback Clock