Application Report

OSPI Controller PHY Tuning Algorithm

Zachary Brown

ABSTRACT

Octal Serial Peripheral Interface (OSPI) is an 8-bit, bidirectional data interface, primarily used for communication with external flash memory devices. This application report explains how to tune the OSPI controller and PHY used on the Jacinto 7 family of SoCs, including the TDA4 and DRA8 SoC variants in 166 MHz DDR mode. It describes the tuning algorithm and why it is needed.

Project collateral and source code discussed in this application report can be downloaded from the following URL: http://www.ti.com/lit/zip/spract2.

Table of Contents

List of Figures

Trademarks

1 Introduction

Figure 1-1 shows the OSPI controller connected to the Flash device.

The data lines (DQ[7..0]) are bidirectional. During the command and address sections of the read transaction the controller drives those pins. During the data section of the transaction the flash devices drives the data lines. Figure 1-2 is an example of 4-byte read transaction.

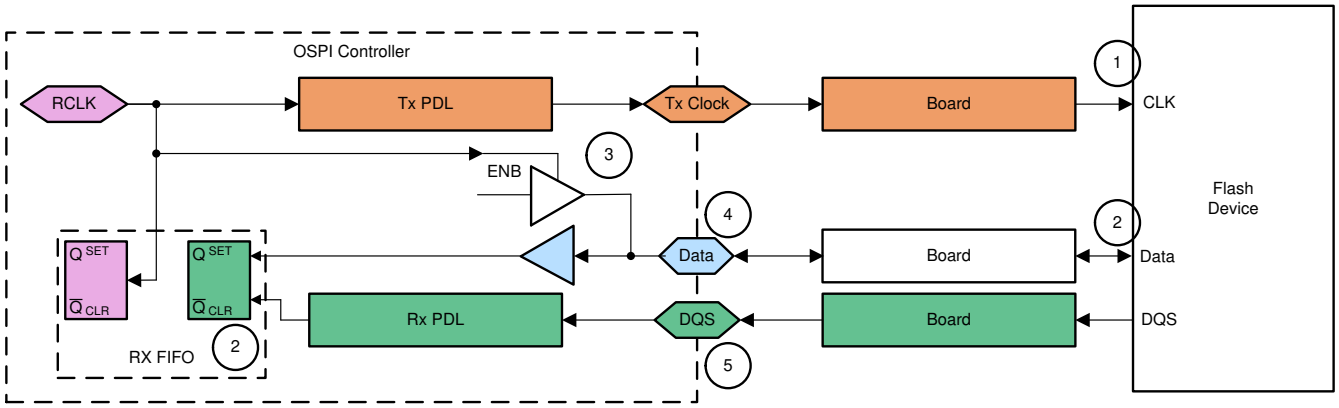

Figure 1-1. OSPI Controller and Flash Device

The controller provides the OSPI clock to the Flash device. It is generated by delaying the ref_clk through the TX PDL. The flash device uses the clock to capture the command and address during the command and address phases. During the data phase, the OSPI device drives a new data byte on each edge of the OSPI clock. Figure 1-2 is an example of 4-byte read transaction.

Some OSPI devices provide a DQS signal. The DQS and data are edge aligned at points 4 and 5 in Figure 1-1. DQS must be delayed by the RX PDL to a point inside the data eye to sample valid data at point 2.

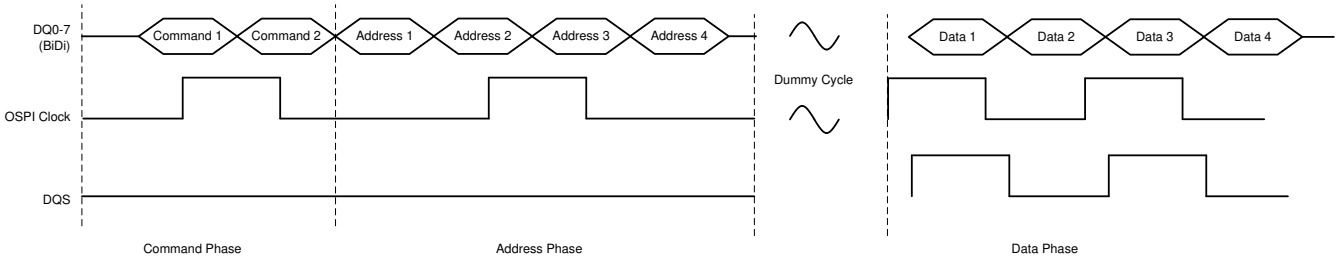

Figure 1-2. Timing Diagram of a Read Transaction

The “round trip delay” of data is the time from a ref_clk edge, to the sampling time of the data triggered by that edge. The sum of delays created by the TX PDL, the travel time of the OSPI clock from the controller to the flash device, the output delay of the flash device, and the RX PDL Delay, creates the round trip delay. The controller samples the data into an RX FIFO using the Delayed DQS. The data is read by the controller out of the RX FIFO using the ref_clk.

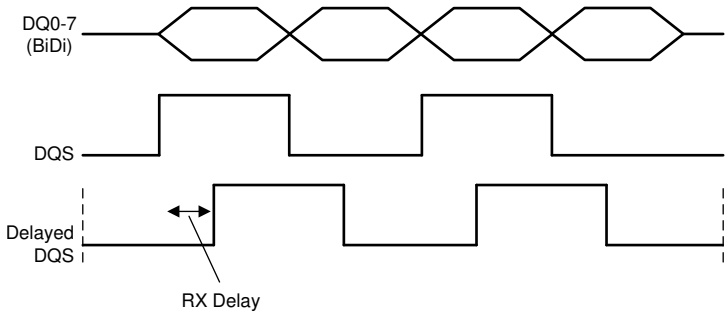

Figure 1-3. Data Sampling

The controller expects the first byte of data to be captured within a specific ref_clk cycle (the target cycle), and all remaining data in the following cycles. In cases where the round trip delay is higher than the ref_clk period, the target cycle must be moved to the next ref_clk cycle using the Read Data Capture register's Read Delay field OSPI_RD_DATA_CAPTURE_REG[4:1].

The goal of the tuning procedure is to select an optimal tuning point (OTP) of Read Delay, TX PDL Delay, and RX PDL Delay for sampling data.

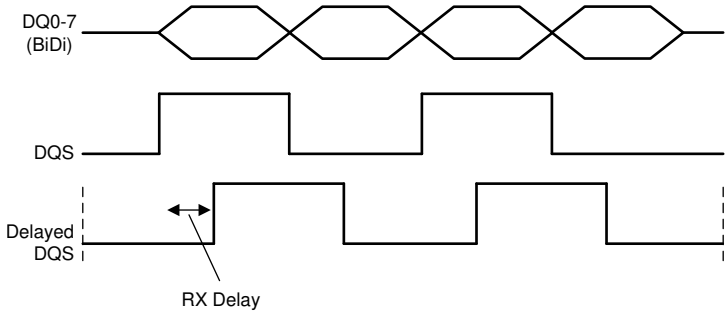

Figure 1-4. RCLK Target Cycles

1.1 Acronyms Used in This Document

Table 1-1. Acronyms Used in This Document

Acronym	Description
Ref_clk	The internal clock of the OSPI controller.
OSPI Clock	The clock of the OSPI bus.
DQS	Sometimes referred to as data strobe, this is a signal provided by some OSPI devices. It acts as a high speed clock for the data lanes. The controller can use a delayed DQS to sample incoming data.
DLL	Delay locked loop
PDL	Programmable delay line
OSPI PHY	The part of the OSPI controller which sets up TX delay, and samples incoming data.
Read Delay	A parameter of the OSPI controller which determines which ref_clk cycle incoming data must be sampled in.
Data Eye	The period of time in which all data bits are valid. The sampling edge must occur inside the data eye for the byte to be read successfully.
OTP	Optimal Tuning Point

2 Tuning Algorithm

2.1 Passing Regions

All combinations of TX and RX delay values can be viewed as a 2 dimensional plot, with RX PDL Delay on the horizontal axis, and TX PDL Delay on the vertical axis. Figure 2-1is a stylized, representative plot showing typical TX, RX, and Read Delay configurations that will allow the OSPI PHY to read successfully. Colored areas show the TX and RX combinations for the different ref_clk targets that result in valid reads (passing region). White space represents TX and RX combinations in which valid data will not be read (failing regions).

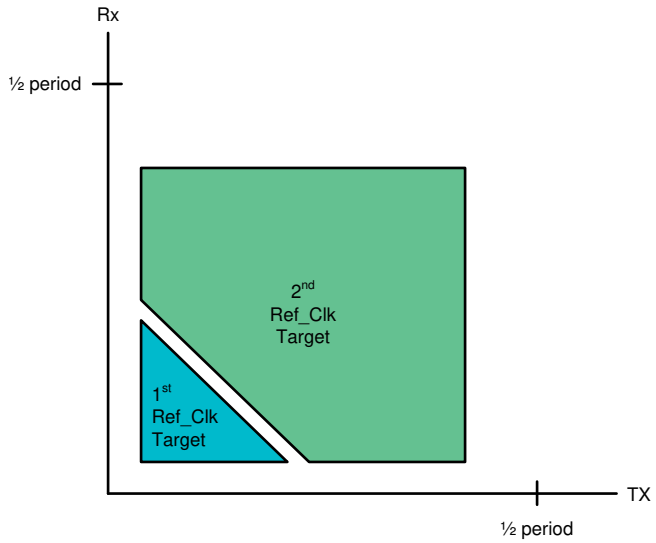

Figure 2-1. TX and RX Passing Regions

The passing region is divided into two sub regions, each one corresponding to a target cycle. The OSPI tuning algorithm identifies the largest region, selects that corresponding ref_clk target, and sets the TX and RX PDL delays to sample within that ref_clk target.

TX min and max (side walls of the passing region) are formed by the setup and hold time requirement of the OSPI device. TX delays outside this range cause command and address bytes to be latched incorrectly by the OSPI device, resulting in an unsuccessful read.

RX min and max (top and bottom of the passing region) are formed by the setup and hold time requirement of OSPI Controller. RX delays outside this range cause data bytes to be latched incorrectly by the OSPI controller, resulting in an unsuccessful read.

Both the TX PDL Delay and RX PDL Delay contribute to the round trip delay, which pushes the sample point from one ref_clk cycle to the next. The diagonal line between sub regions exists because the sum of PDL delays must not exceed a fixed value in order to sample within the 1st Ref_Clk Target.

2.2 Temperature Effect on Passing Region

SoC die temperature affects the IO delays, causing changes in the size and location of the passing regions, as shown in Figure 2-2.

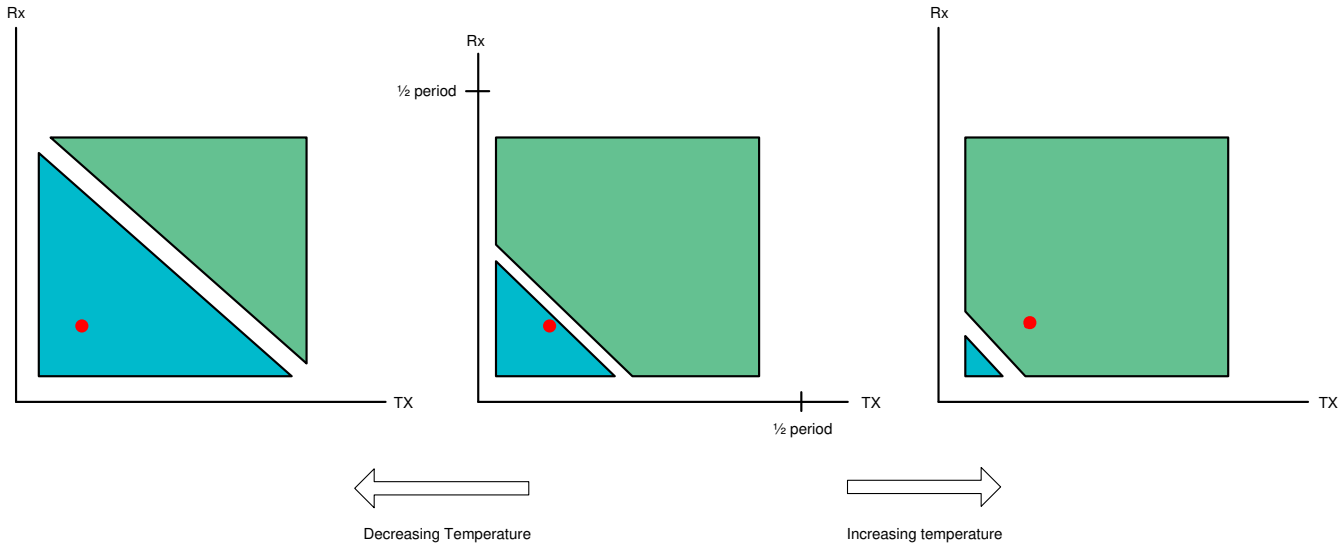

Figure 2-2. Temperature Effect on Passing Regions

The boundary between the two passing regions moves with temperature. If that boundary shifts during operation and crosses over the selected tuning point, reads from the OSPI controller will fail to read correct data. Figure 2-2 shows an example of a poorly placed tuning point. The plot on the left shows tuning at low temperature. As temperature increases, IO delays increase, causing the round trip delay to increase. TX and RX PDL values remain the same, and as a result, at high temperatures the sampling point will be in the wrong ref_clk cycle. To account for this, the OSPI tuning algorithm identifies the largest passing region, and selects a TX/RX combination far away from the moving boundary between ref_clk cycles.

2.3 Algorithm

The OSPI tuning algorithm tunes the PHY by identifying the key features of the passing region. These key features are:

- Bottom Left Corner – The tuning algorithm fixes TX and searches RX to find the RX Min and RX Max. Then it fixes RX and searches for TX Min and TX Max. The Bottom Left corner is identified at the point TX Min, RX Min.
- Top Right Corner – The Top Right corner is identified at the point TX Max, RX Max.
- Location of Read Delay Boundary – The tuning algorithm executes a binary search along the line between points 2 and 3, which is shown in Figure 2-3 as an arrow.

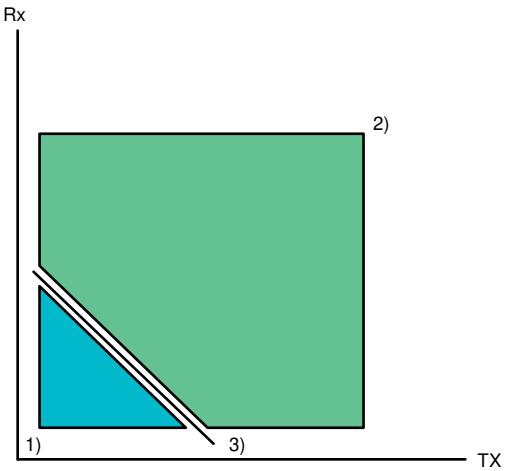

Figure 2-3. Tuning Algorithm Search Points

In order to test whether a TX/RX/Read Delay combination passes, the tuning algorithm configures the PHY to that setting, and reads a known pattern from the OSPI memory. The test pattern can be found in the associated zip file: http://www.ti.com/lit/zip/spract2. The tuning algorithm searches along the line between points (1) and (2), testing TX/RX combinations between them, and identifies the position of the boundary represented by line (3). The algorithm then identifies which ref_clk target is associated with the largest passing region, and sets TX/RX delays into the corresponding corner, leaving some margin for the TX/RX minimum and maximum to shift. Figure 2-4 shows the placement of the OTP in two scenarios, based on which passing region is larger.

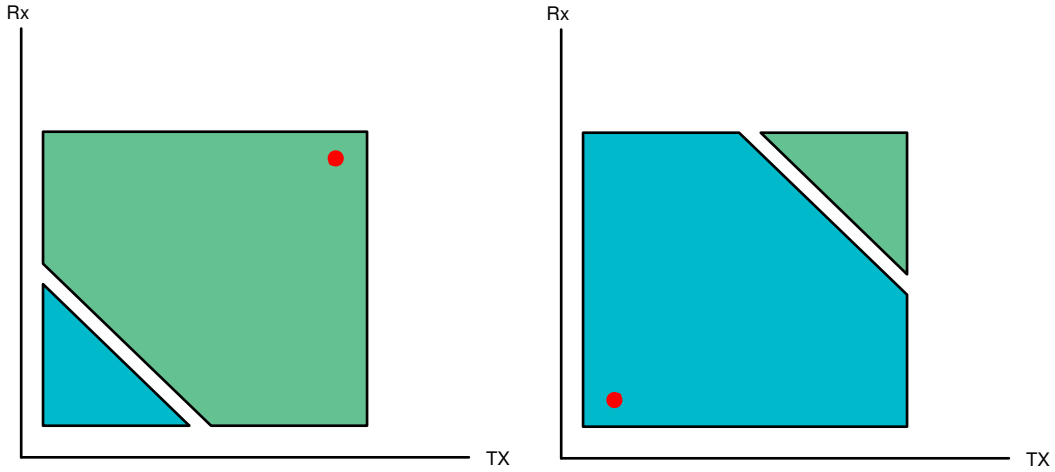

Figure 2-4. OTP Selection When Two Passing Regions are Present

If only one tuning window is detected, the OTP is placed according to temperature, as shown in Figure 2-5.

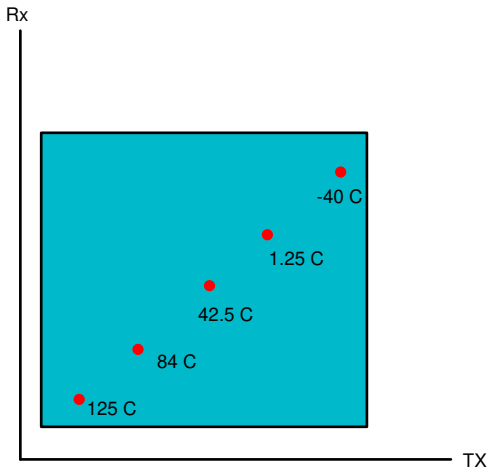

Figure 2-5. OTP Selection When One Passing Region is Present

For implementation details of the algorithm, see https://git.ti.com/cgit/ti-linux-kernel/ti-linux-kernel/tree/drivers/spi/spi-cadence-quadspi.c?h=ti-linux-5.4.y. Information on tuning the PHY in SDR mode will be included in later versions of this document. Considerations for using this algorithm at frequencies other than 166 MHz will also be addressed.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265