Implementing Video Display Interfaces Using MachXO2 PLDs

A Lattice Semiconductor White Paper

November 2010

Lattice Semiconductor
5555 Northeast Moore Ct.
Hillsboro, Oregon 97124 USA
Telephone: (503) 268-8000
www.latticesemi.com

Introduction

Lattice Semiconductor has developed and supported a number of reference designs for display interfaces in various devices. A display interface is now available in the MachXO2 PLD family. Because this interface is now supported in MachXO2 devices, designers have an even lower cost and very low power alternative to implement embedded displays. As described in this white paper, the MachXO2 provides a compelling choice for this application.

The display interface is a 7:1 LVDS interface. In the following sections, we will provide an overview of this display standard and explain how key hardware features in MachXO2 devices enable this implementation. In addition, the details of available reference designs and relevant application examples will be reviewed.

Overview of Display Interface (7:1 LVDS)

Applications which require an integrated LCD screen will typically use a display interface (7:1 LVDS) connection. This interface is also commonly referred to as Channel Link, Flatlink or Camera Link from other vendors. Technically, Channel Link and Flatlink are specified for LCD displays and Camera Link is targeted at digital cameras and frame grabbers. Regardless of the name, this 7:1 interface uses a LVDS differential signaling I/O standard. The VESA (Video Electronics Standards Association) defined the 7:1 LVDS interface for LCD displays. It has become very popular in laptops and netbooks to attach the motherboard to the LCD screen. Today a number of different pinout interfaces from 20 – 50 pins exist to support various LCD display resolutions. See Figure 1 below for an example of 7:1 cabling being used in a laptop.

Figure 1 – 7:1 LVDS Cabling for Connecting LCD Screen to Laptop Motherboard

To facilitate the transmission of digital display data to an LCD screen and minimize connections, the data that is being sent from a motherboard is converted to a serial format. The display interface typically uses either three or four LVDS data lanes and one LVDS clock lane. Higher resolution displays would use four LVDS data lanes and one LVDS clock. In one clock period or cycle, there are 7 serial bits on each data lane, as illustrated in Figure 1.

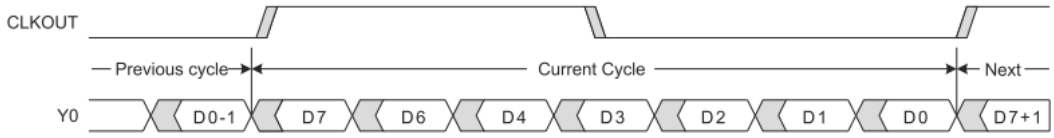

Figure 1- Embedded Display Interface (7:1 LVDS) Timing

Thus in the LVDS display Interface, the data rate is 7 times faster than the clock period. The one clock lane typically supports three or four data lanes. Therefore at the display either 21 or 28 bits of data are received. For reference, the clock lane typically operates from a rate of 60-100Mhz.

Each pixel of a LCD display has red, green and blue elements that make up all the possible colors for the screen. If three LVDS data lanes are being transmitted, for example from a laptop motherboard, then the colors will contain 6 data bits of RGB (Red, Green, and Blue) data plus a control bit. These bits are then serialized into LVDS differential lanes. See table 1 for a representation of how three lanes and their 21 serial bits make up the RGB data. A similar relationship exists for the four lane interface, which is 28 bits. In a four lane interface 8 bits of RGB color are utilized.

Serial Channel	Data Bits	Format
Y0	D0	Red0
	D1	Red1
	D2	Red2
	D3	Red3
	D4	Red4
	D6	Red5
	D7	Green0
Y1	D8	Green1
	D9	Green2
	D12	Green3
	D13	Green4
	D14	Green5
	D15	Blue0
	D18	Blue1
Y2	D19	Blue2
	D20	Blue3
	D21	Blue4
	D22	Blue5
	D24	Hsync
	D25	Vsync
	D26	Enable

Table 1- Three Lane 7:1 LVDS Data

Typically the transmitter will send the serial lanes to the receiver device, such as an LCD screen. The LCD screen will then de-serialize these three or four LVDS lanes. As previously mentioned, the serial signals are for Red, Green, Blue, as well as the control bits. When four LVDS lanes are used to transmit, then each RGB pixel will have 8 bits of data and the associated control bits. Whether there are 21 or 28 bits of video data,

the LCD uses this RGB data and drives the appropriate image to the screen. For further details on the Display Interface (7:1 LVDS interface, Flatlink, Camera Link or Channel Link), please refer to the Reference Design document RD1093: MachXO2 Display Interface.

MachXO2 Features Support Display Interfaces (7:1 LVDS)

There are several design techniques that can be used to implement the display interface. The general approach is to receive the incoming clock and multiply it by 7 times to clock each bit of data. In reality, this is quite difficult because the clock has to run extremely fast. Since a typical display interface clock rate is 60-100Mhz or higher, multiplying this by 7 yields a frequency of 420-700Mhz. Given that a printed circuit board adds trace delay variation, capacitance, noise, etc. this interface is a significant challenge to design.

The MachXO2 has been architected with specific features to support the display interface. These are double data rate (DDR) I/O registers, I/O register gearing logic and dedicated clock divider by 3.5. In fact, the MachXO2 has specific 7:1 input and output I/O banks and software macros to make this implementation straightforward. In the MachXO2-1200 and larger devices, display interface inputs are supported on the bottom I/O banks and display interface outputs are supported on the top I/O banks. Let's examine each architecture feature in further detail.

The DDR I/O registers in the MachXO2 allow the display interface to be clocked at half the rate of the interface data speed. For example, if the display interface pixel clock rate is 100Mhz, using the DDR I/O registers allows a clock to run at 350Mhz using both edges, instead of a 700Mhz clock. A PLL in the device multiplies the clock by 3.5 and both edges are used to clock the data lanes. This reduced clock rate allows higher throughput speeds to be supported. The MachXO2 can support a maximum clock rate of up to 108Mhz or throughput of 756Mbps.

The I/O register gearing logic performs the task of taking the output of the DDR I/O and time division multiplexing it to a wider bus. Working with the DDR I/Os, the output of the register gearing logic is a 7 bit wide data bus that was generated from one data lane

running 7 times faster. For example if the one data lane was running at 560Mbps,(7 x 80Mhz) then the output of the register gearing logic will time division multiplex this to 7 bit running at 80Mhz. If the display interface has four data lanes, then at the output of the register gearing logic will be 28 internal data bits running at the same speed as the external clock. See figure 2 for a diagram of the DDR I/O registers and the I/O register gearing logic.

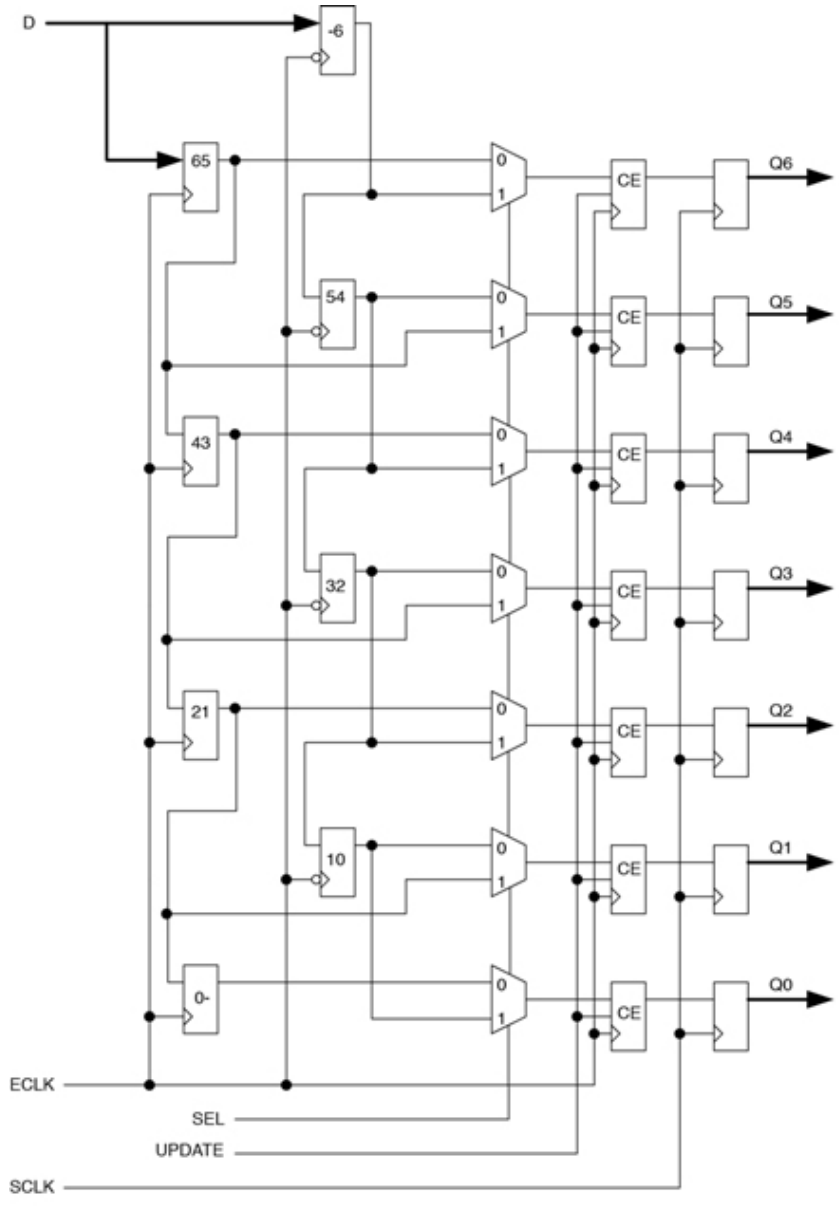

Figure 2 – DDR I/O Registers and I/O Register Gearing Logic

The dedicated clock divider by 3.5 precisely converts the multiplied up input clock so the output of the I/O register gearing is reliably clocked into the FPGA fabric. Because of this divider, the I/O gearing can be guaranteed to transfer the data from the I/O to the MachXO2 PLD fabric. Although it may seem odd to use a PLL on the input clock to multiply up by 3.5 only to have the dedicated clock divider take it back down by 3.5, there are benefits to doing this. If this dedicated divider was not present, the raw input clock or another PLL output would be fed to the output of the I/O register gearing logic. One would have to carefully hand route this clock to ensure there was minimal skew or variation delay across the 21-28 data bits. In addition, the interface speed would be limited by this clock skew. The dedicated divider also does not consume any other system resources, such as another PLL output. With the dedicated divider, the clock domain transfer timing and interface to the MachXO2 fabric from the I/O gearing logic is guaranteed. For further details on the I/O features that support the Display interface, see TN1023 (MachXO2 High-Speed Source Synchronous and Memory Interfaces.)

Display Interface Design Examples

Lattice offers both Transmit (Tx) and Receive (Rx) functionality for the Display Interface in a broad range of FPGA devices in addition to the MachXO2 (Please refer to RD1030 – LatticeECP3, LatticeECP2/M, LatticeXP2 7:1 LVDS Video Interface Reference Design). As a number of Lattice devices can be used to implement the 7:1 LVDS, a wide variety of display applications can be supported in embedded or consumer designs. The Display Interface (7:1 LVDS interface) in the MachXO2 family makes it ideal for battery powered products and all lower cost products such as a laptop computers, netbooks etc. The MachXO2 is a low cost and low power solution for implementing a display interface.

One display interface application which is often found in a notebook is a graphics multiplexer controller, or GMUX. A GMUX simply exists to select one of the two graphics controllers to drive the LCD screen. This function is most common in laptops with higher resolution screens. These laptops have two video sources which could drive the LCD screen. The two graphics controllers share the single LCD screen. Usually one controller is optimized or built into a processor chip set for low power

consumption and less demanding tasks such as word processing. The second graphics controller is designed for higher performance and more intensive display applications such as gaming or video editing. See figure 3 for further details.

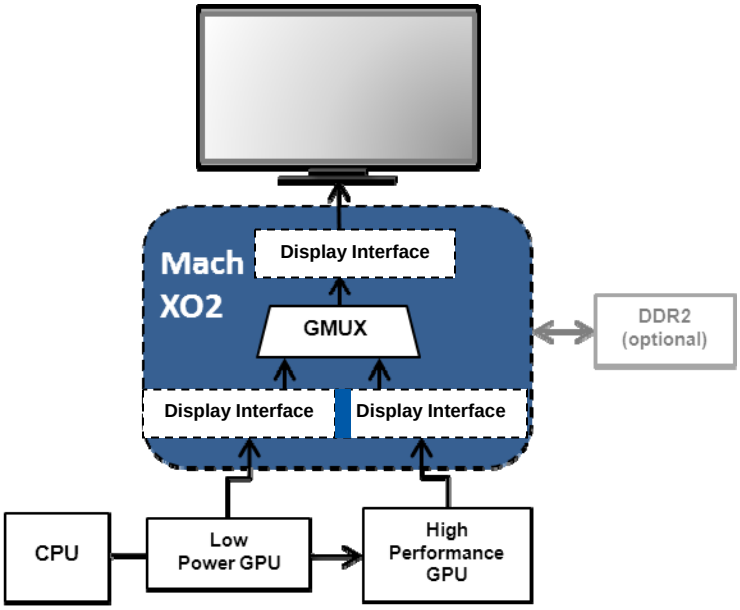

Figure 3- Graphics Multiplexer Controller (GMUX)

By implementing two display interface Rx interfaces and muxing them to a display interface Tx output in a MachXO2 device, a notebook can achieve several of the following benefits:

1. A discrete graphics multiplexer unshackles a PC vendor from having to use a particular company's graphics switching standard. Not all processors and GPUs are designed to seamlessly work together. For example if a particular Intel microprocessor is the best CPU selection and a high performance AMD GPU is most compelling for graphics, a GMUX function facilitates this marriage.
2. Second, the decision of when and how to switch to which graphics processor can also be flexible. A laptop could use the higher performance GPU when it detects that an HDMI cable has been plugged in and then revert to the lower power GPU when it is removed. Alternatively, the switching could be implemented on the fly if the display data is stored in a small buffer, such as a DDR2 memory. This ensures that switching will be undetectable to the user and no system reboot will be required.

Lastly, the discrete GMUX will also allow a system to power down the graphics controller that is not being used. Some automatic switching graphic schemes in laptops actually never power off the integrated controller. They inefficiently power-up the higher performance GPU and over write the frame buffer data from the integrated graphics controller. By using a low power MachXO2 device for the GMUX function, you can power off either graphics processor thus conserving battery life to the fullest extent possible.

Summary

This white paper has provided a detailed overview of the display interface, and the MachXO2 device's capability to implement this interface as well as design examples. The abundant architectural features of the MachXO2 device combined with its ultra low power and low cost make it a compelling choice for implementation of the display interface (7:1 LVDS). The table below lists all of the Lattice device families that can be utilized as well as the available documentation to support this application.

Supported Device Families for 7:1 LVDS			
MachXO2	**ECP3**	**ECP2M**	**XP2**
TN1203 RD1093	RD1030 TN1134 UG37	RD1030 TN1134 UG37	RD1030 TN1134 UG37

TN1023 – MachXO2 High-Speed Source Synchronous and Memory Interfaces Tech Note

RD1093 –MachXO2 Design Interface Reference Design

RD1030 – LatticeECP3, LatticeECP2/M, LatticeXP2 7:1 LVDS Video Interface Reference Design

TN1134 - 7:1 LVDS to DVI Tech Note

UG37 – Users Guide for 7:1 LVDS to DVI demo